

**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC**

**MÉMOIRE PRÉSENTÉ À
L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE**

**COMME EXIGENCE PARTIELLE
À L'OBTENTION DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE
M.Ing.**

**PAR
HUSSEIN HASSAN**

**TÉLÉMANIPULATION SYNCHRONE ET COMMANDE
D'INSTRUMENTS PHYSIQUES DANS LE CADRE
D'UN LABORATOIRE VIRTUEL**

MONTREAL , LE 16 JANVIER 2002

© droits réservés de Hussein Hassan

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

- **M. Maarouf Saad, directeur de mémoire**
Département de génie de la production automatisée
- **M. Michel Lavoie, président du jury**
Département de génie électrique
- **M Hamadou Saliah**
Science et technologie, Télé-Université
- **M. Mohamed Cheriet**
Département de génie de la production automatisée

**IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET UN
PUBLIC**

LE 7 NOVEMBRE 2001

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

TÉLÉMANIPULATION SYNCHRONE ET COMMANDE D'INSTRUMENTS PHYSIQUES DANS LE CADRE D'UN LABORATOIRE VIRTUEL

Hussein Hassan

SOMMAIRE

Dans la grande vague de virtualité qui déferle sur notre réalité, un nouveau venu vient enrichir l'entreprise virtuelle, c'est le Laboratoire Virtuel. Ce dernier permet de manipuler, par les réseaux informatiques, des équipements scientifiques pour la réalisation d'expériences à distance. Les étudiants auront alors plus facilement accès à ces nouveaux moyens d'apprentissage, ils peuvent contrôler des équipements, à l'aide d'un micro-ordinateur relié au réseau pour réaliser, en temps réel et à distance, des séances de laboratoire à partir de leur domicile. Ces nouveaux types d'environnements d'apprentissage offrent des ressources et des outils permettant d'atteindre des objectifs d'apprentissage comparables à ceux qui sont visés par les laboratoires réels, c'est-à-dire en présentiel.

Mon projet de maîtrise, qui représente une partie d'un projet d'envergure sur les laboratoires virtuels, consiste à éclaircir l'idée du laboratoire virtuel en général, à la mise au point de méthodes permettant d'utiliser à distance un ensemble de matériels ou d'instruments de mesure et à explorer un ensemble d'outils qui peuvent être employés dans la mise en œuvre de laboratoires virtuels.

Dans ce mémoire, nous présenterons les éléments du laboratoire virtuel avec leurs définitions et leurs rôles. Nous ferons aussi une étude sur la technique de communications entre un micro-ordinateur et des instruments physiques ainsi que les outils nécessaires. Nous montrerons l'accès à distance d'un moteur à courant continu et d'un système simulant des procédés permettant un contrôle de débit, de pression ou de niveau.

ABSTRACT

In distance education, laboratory experimentation is inconvenient because the students usually have to be physically present in the universities laboratories. A solution to avoid the disadvantages of local experimentation is the tele-operation of laboratory experiments. Providing remotely accessible experiments, unique or expensive equipment can be shared between several universities.

Our project aims to create a collaborative learning environment for a remote laboratory that allows students to manipulate instruments, conduct experiments and view image from their home. Students will have access to the remote laboratory via Internet from any where at any time, they will control the experiments exclusively with their standard web browser, and they will be able to interact and discuss the results of their work at the same time, in the future, they don't have to be physically present in theirs laboratories.

A remote laboratory offers a cost effective and flexible means for students to share access to physical experiments. It is based on a client server architecture, which is mainly implemented with LabView programming language and using TCP/IP protocol.

The student may be either a controlling client or a monitoring client. A controlling client can start and stop the experiment, adjust parameters and collect data. A monitoring client, who may be the instructor or other student, can only observe what the controlling client is doing.

This paper describes the development and implementation of methods in order to set up a prototype of an experimental environment, presents the technical structure and shows results of the project. Interface electronics and control software are also discussed.

REMERCIEMENTS

Je tiens tout d'abord à exprimer ma reconnaissance à mon directeur Dr Maarouf Saad pour la qualité de son encadrement, son soutien inappréciable, ses idées et ses conseils durant toute la période du projet, ainsi que son appui moral.

J'aimerais aussi remercier sincèrement Dr Hamadou Saliah, professeur et coordinateur du projet Synchronédia, pour l'aide qu'il m'a apporté afin de réaliser mon projet.

J'exprime ma reconnaissance envers Dr Mohamed Chériet pour ses idées et son appui qui m'ont permis de terminer mon travail.

Mes remerciements s'adressent également à tous ceux qui ont contribué à la réalisation de ce travail et particulièrement Pauline et tous les gens du laboratoire où j'ai travaillé pour la sympathie qu'ils ont manifestée et pour leur collaboration.

Je ne dois pas oublier de remercier Nadia et ma petite fille Sarah qui a pu supporter mon absence durant mes études, j'espère qu'elle comprendra un jour et suivra le même chemin.

Enfin, je suis très reconnaissant envers Dieu de m'avoir donné la volonté et le courage afin de réaliser mon travail.

TABLE DES MATIÈRES

	Page
SOMMAIRE	i
ABSTRACT	ii
REMERCIEMENTS	iii
TABLE DES MATIÈRES	v
LISTE DES FIGURES	viii
LISTE DES ABRÉVIATIONS ET DES SIGLES	xii
 INTRODUCTION	 1
 CHAPITRE 1 : REVUE DE LA LITTÉRATURE	 5
 CHAPITRE 2 : LABORATOIRE VIRTUEL : FORMATION ET ASSISTANCE TECHNIQUE MÉDIATISÉE	 12
2.1 Introduction	12
2.2 Buts	12
2.3 Définitions	14
2.3.1 Laboratoire	14
2.3.2 Virtuel	14
2.3.3 Laboratoire virtuel	14
2.3.4 Apprenant	15
2.3.5 Formateur	16
2.3.6 Administrateur ou gestionnaire	16
2.4 Potentiel (en dimension culturelle)	16
2.5 Aspects pédagogiques	17
2.6 Aspects techniques	19
2.7 Outils et approches	21
2.7.1 L'Internet	26
2.7.2 Le débit des réseaux	27
2.7.3 L'acquisition de données	27
2.8 La carte d'acquisition	28
2.8.1 Généralisation	28
2.8.2 Canaux différentiels vs Asymétriques	29
2.8.3 Convertisseur A/N	30

2.8.4 Débit.....	31
2.8.5 Amplificateur d'entrée.....	32
2.8.6 Entrées/sorties numériques	32
2.8.7 Compteurs et temporisateurs.....	32
2.9 Les réseaux.....	33
2.9.1 Ethernet	35
2.9.2 VLAN	36
2.9.3 VPN.....	36
2.10 Protocole de transmission	37
2.10.1 Le protocole UDP	37
2.10.2 Le protocole TCP/IP (<i>Transmission Control Protocol/Internet Protocol</i>).....	37
2.11 Le protocole GPIB	39
2.12 Logiciels.....	42
2.13 Problèmes.....	44
2.14 Avantages et inconvénients.....	45
2.15 Statistiques	46
2.16 Conclusion	47

CHAPITRE 3 : COMMANDE À DISTANCE D'UN MOTEUR À COURANT CONTINU EN TEMPS RÉEL

3.1 Introduction.....	48
3.2 Généralités	49
3.3 Identification et Modélisation des systèmes dynamiques.....	50
3.3.1 Notion de Système	51
3.3.2 Processus (procédé).....	52
3.3.3 Modélisation d'un système	52
3.4 Modèles mathématiques.....	53
3.4.1 Classification des modèles mathématiques.....	54
3.5 Construire un modèle.....	55
3.5.1 Modélisation physique (modèle de connaissance).....	55
3.5.2 Identification (modèle de comportement ou boîte noire)	55
3.6 Asservissement et régulation	56
3.6.1 Asservissement	56
3.6.2 Régulation	57
3.7 Qualités d'un asservissement.....	58
3.8 Contrôleur (Régulateur).....	59
3.8.1 Régulateur PID.....	60
3.9 Boucle de contrôle	64
3.9.1 Boucle de contrôle ouverte	64
3.9.2 Boucle de contrôle fermée	64
3.10 Moteur à courant continu	65
3.10.1 Les éléments constitutifs du moteur.....	66
3.10.2 Le fonctionnement du moteur.....	67
3.10.3 Construction du schéma-bloc.....	72

3.11 L'encodeur	74
3.12 Les éléments du système.....	75
3.13 Réalisation.....	79
3.14 Identification des paramètres	84
3.15 Simulation.....	87
3.15.1 En boucle ouverte	87
3.15.2 En boucle fermée	88
3.15.3 Conception du P.D.	90
3.16 Résultat avec LabVIEW.....	94
3.17 Le contrôle à distance en utilisant le protocole TCP/IP	99
3.17.1 Algorithme d'une application client - serveur	102
3.17.2 Time-Out et erreurs.....	104
3.17.3 Applications réalisées	105
3.17.4 Contrôle du moteur à distance	110
3.18 Conclusion	112
 CHAPITRE 4 : COMMANDE À DISTANCE D'UN SIMULATEUR DE PROCÉDÉS	
.....	114
4.1 Introduction.....	114
4.2 Description du simulateur	114
4.3 Architecture du laboratoire virtuel.....	119
4.4 Communication avec les ports RS232 et RS485	122
4.5 Application.....	123
4.6 Conclusion	126
 CONCLUSION.....	127
 ANNEXES	
A : LABVIEW	129
B : L'encodeur incrémental.....	139
C : Moteur et encodeur	145
D : Schéma interne de la carte d'acquisition.....	147
E : Les E/S connexions pour la carte PCI-MOI-16 E-4	149
F : Les programmes avec Matlab.....	151
G : Code Interface Node (CIN).....	156
H : Moteur avec le banc d'essai	169
I : Les diagrammes de LabVIEW	171
 BIBLIOGRAPHIES.....	185

LISTE DES FIGURES

	Page
Figure 2. 1 : Composants du laboratoire virtuel.....	21
Figure 2. 2 : Un modèle d'un environnement virtuel.....	25
Figure 2. 3 : Une carte d'acquisition pour lire des signaux.	29
Figure 2. 4 : Réseaux LAN.	34
Figure 2. 5 : Ethernet sur LAN (Local Area Network).	35
Figure 2. 6 : Le concept IEEE 488, contrôleur et périphériques.....	41
Figure 3. 1 : Représentation générale.	49
Figure 3. 2 : Notions d'un système dynamique.	50
Figure 3. 3 : Effets de perturbations.....	52
Figure 3. 4 : Contraintes prises en compte dans la synthèse.....	56
Figure 3. 5 : Système de commande.	58
Figure 3. 6 : Représentation générale d'un régulateur.....	59
Figure 3. 7 : Régulateur PID.	60
Figure 3. 8 : Rôle des actions P, I et D.	61
Figure 3. 9 : Régulateur parallèle avec référence variable.....	63
Figure 3. 10 : Régulateur parallèle avec référence constante	63
Figure 3. 11 : Régulateur mixte.	63
Figure 3. 12 : Système boucle ouverte.....	64
Figure 3. 13 : Système boucle fermée.....	65
Figure 3. 14 : Conversion d'énergie.	65
Figure 3. 15 : Symbole.....	65

Figure 3. 16 : Un moteur.....	66
Figure 3. 17 : Schéma d'un moteur à CC.	68
Figure 3. 18 : Schéma de trois phénomènes dans un moteur à CC.....	69
Figure 3. 19 : Phénomène électrique au moteur à CC.	69
Figure 3. 20 : Schéma bloc de système électrique.	71
Figure 3. 21 : Schéma bloc de système magnétique.	71
Figure 3. 22 : Schéma bloc de système mécanique.....	71
Figure 3. 23 : Schéma bloc du moteur c.c.....	72
Figure 3. 24 : Le système au complet.	75
Figure 3. 25 : Interface entre PC et moteur.....	76
Figure 3. 26 : Un signal TTL	77
Figure 3. 27 : Sous palettes pour l'acquisition de données.....	78
Figure 3. 28 : Voies configurées.	79
Figure 3. 29 : Voie analogique pour commander le moteur.	80
Figure 3. 30 : Configuration d'une voie analogique.	80
Figure 3. 31 : Schéma simplifié d'un compteur.....	81
Figure 3. 32 : Compter les impulsions de l'encodeur.	81
Figure 3. 33 : Lire la position.....	82
Figure 3. 34 : les signaux A et B de l'encodeur.....	82
Figure 3. 35 : Le schéma de connexions.....	83
Figure 3. 36 : Contrôle avec PID.	83
Figure 3. 37 : Interface d'identification.	85
Figure 3. 38 : Position et vitesse du moteur.....	86
Figure 3. 39 : Position en boucle ouverte (Matlab).....	87
Figure 3. 40 : Vitesse en boucle ouverte (Matlab).....	88
Figure 3. 41 : Position en boucle fermée sans PD (Simulink).	89
Figure 3. 42 : Vitesse en boucle fermée sans PD (Simulink).	89
Figure 3. 43 : Commande de la position par régulateur PD.....	90
Figure 3. 44 : Vérification avec Matlab.	92

Figure 3. 45 : Position en boucle fermée avec un régulateur PD.....	93
Figure 3. 46 : Calcul du PID.	94
Figure 3. 47 : La simulation avec LabVIEW.....	95
Figure 3. 48 : L'interface du contrôle en temps réel.....	96
Figure 3. 49 : Algorithme suivi pour contrôler le moteur en temps réel.....	97
Figure 3. 50 : Résultats sans PID.	98
Figure 3. 51 : Résultats avec PID.....	99
Figure 3. 52 : Les fonctions TCP/IP avec LabVIEW.	100
Figure 3. 53 : Serveur qui écrit dans le client.	105
Figure 3. 54 : Client qui reçoit du serveur.	106
Figure 3. 55 : Serveur reçoit du client.....	107
Figure 3. 56 : Client envoie vers le serveur.	107
Figure 3. 57 : Serveur reçoit puis il renvoie vers le client.	108
Figure 3. 58 : Client qui envoie puis il reçoit du serveur.....	109
Figure 3. 59 : Application serveur qui contrôle le moteur.....	110
Figure 3. 60 : L'application client qui permet de contrôler le moteur à distance.	111
Figure 3. 61 : Application client serveur.....	112
Figure 4. 1 : Principe de fonctionnement du transmetteur.....	115
Figure 4. 2 : Les ajustements interne et externe du transmetteur.....	116
Figure 4. 3 : Montage de calibrage des transmetteurs de pression.	117
Figure 4. 4 : Simulateur de procédés.	118
Figure 4. 5 : Configuration locale	119
Figure 4. 6 : Simulateur de procédés vu par la caméra PROSHARE.	120
Figure 4. 7 : Interface LabVIEW du simulateur.	121
Figure 4. 8 : Illustration de la communication RS485.....	122
Figure 4. 9 : Vue d'ensemble du simulateur.....	124
Figure 4. 10 : Réponse à l'échelon.....	125
Figure A. 1 : La programmation avec LabVIEW.	131
Figure A. 2 : Interface avec LabVIEW.....	133

Figure A. 3 : La face diagramme.	134
Figure A. 4 : la palette d'outils.	135
Figure A. 5 : la palette de commandes.	136
Figure A. 6 : La palette de fonctions.	137
Figure A. 7 : Les flux de données dans le diagramme LabVIEW.	138
Figure B. 1: Encodeur incrémental.	140
Figure B. 2 : Les impulsions dans un encodeur incrémental.	141
Figure B. 3 : Déphasage entre les deux signaux A et B.	142
Figure B. 4 : Sens de rotation.	142
Figure B. 5 : Précision.	143
Figure B. 6 : Parasites.	144
Figure C. 1 : Moteur et encodeur.	146
Figure D. 1 : Schéma interne de la carte d'acquisition PCI-MOI-16 E-4.	148
Figure E. 1 : Les E/S connexions pour la carte d'acquisition.	150
Figure F. 1 : Boucle fermée avec Simulink.	152
Figure G. 1 : CIN avec ces paramètres.	159
Figure G. 2 : Création d'un nouveau projet.	162
Figure G. 3 : Les fichiers qu'on aura besoin dans le projet.	163
Figure G. 4 : Configuration pour le projet.	164
Figure G. 5 : Configuration pour Code Generation.	165
Figure G. 6 : Le chemin d'accès.	166
Figure H. 1 : Le banc d'essai avec le moteur.	170

LISTE DES ABRÉVIATIONS ET DES SIGLES

LabVIEW	Laboratory Virtual Instrument Engineering Workbench
Hz	Hertz (seconde ⁻¹)
PID	proportional, integral and derivative
SP	Set Point (Signal de consigne)
Réf	Représente la consigne en volts.
PV	Représente la variable du procédé en volts
ISOC	Internet Society
CNRS	Centre National de la Recherche Scientifique
LBNL	Lawrence Berkeley National Laboratory.
LSB	Least significant bit
PC	Personal computer.
GPIB	General Purpose Interface Bus
G	Graphic.
ICANN	Internet Corporation for Assigned Names and Numbers
TTL	transistor-transistor logic
A/N	Analogique numérique
N/A	Numérique analogique
WAN	Wide Area Network.
MAN	Metropolitan Area Network.
LAN	Local Area Network
OSI	Open System Interconnection
ISO	International Standard Organization
PARC	Palo Alto Research Center.

CSMA/CD	Carrier Sense Multiple Access/Collision Detection
μs	micro seconde
LANV	Local Area Network Virtual
VPN	Virtual private network
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
IP	Internet Protocol
ARPA	Advanced Research Project Agency
WWW	World Wide Web.
SCPI	Standard command for Programmable instrumentation
TML	Test and Measure Language
NMA	Nouveaux médias.
V	Volts.
A	Ampère
H	Henry
N.m	Newton-mètre
rad/s	Radian par seconde
DEL	Diodes électrolumineuses
E/S	Entrées-Sorties
DAQ	Data Acquisition Carte
PCI-MOI	Peripheral Component Interconnect- Multi Output Input
PnP	Plug and Play
STC	System Timing Controller
NI-PGIA	National Instrument- Programmable Gain Instrumentation Amplifier
VI	Virtuel Instruments
CIN	Code Interface Node
IDE	Integrated Developer Environment
DLL	Dynamic-Link Library

INTRODUCTION

L'apprentissage est né avec l'humanité : dès son existence, l'homme a commencé à apprendre et jusqu'à maintenant il continue à le faire. Son besoin d'apprentissage continu l'a mené à cette ère dite ère informatisée.

L'éducation reste l'un des enjeux majeurs du développement des nouvelles technologies. Plusieurs questions se profilent à l'horizon du 21^e siècle : quelles seront les nouvelles compétences des formateurs et les nouveaux apports au savoir ? Quel sera le métier de l'enseignant dans le futur ?

Les temps changent, le professeur n'est plus seul dépositaire de la connaissance. La relation maître/élève ne se limite plus à celui qui sait et celui qui veut savoir. Internet, *"ce nouveau prof qui sait tout"*, comme le définissait un collégien de 13 ans [1], a bouleversé cette hiérarchie et remis en question le rôle même de l'enseignant. Il n'est plus celui qui transmet mais celui qui aide à chercher, trier et valider des informations surabondantes et immédiates.

Dans un monde où tout devient numérique, accessible à distance et dépendant de la technologie, la formation est devenue aujourd'hui une priorité et prend un nouveau visage, celui qui ouvre de nouvelles perspectives favorise la formation et l'assistance à distance.

Il faut mettre, selon les règles de l'art, des cours en ligne afin de permettre aux enseignants et formateurs de jouer leur nouveau rôle de facilitateurs. Il reste à

convaincre encore un trop grand nombre d'enseignants réfractaires à la formation par l'intermédiaire des technologies de l'information et de la communication.

Mais comment préparer nos futurs professeurs à ces nouvelles fonctions ? Faut-il leur enseigner les technologies de l'information comme une discipline à part entière ou bien intégrer Internet au processus pédagogique?

En effet, les progrès technologiques constants dans le domaine des technologies de l'information et de la communication ont inspiré une nouvelle méthode d'apprentissage dans le but d'ouvrir de nouveaux horizons à la formation, l'apprentissage et le télé travail. L'utilisation de l'ordinateur et du réseau informatique comme support de formation et d'apprentissage va constituer un virage pour les nouvelles applications scientifiques et techniques. De nouvelles manières de former vont prendre place : les mesures qui nécessitaient la présence des utilisateurs et/ou des équipements pourraient être réalisées à travers les réseaux et les micro-ordinateurs équipés de divers outils d'interface et/ou des périphériques. La différence entre réalité et virtualité est maintenant l'objet de la formation à distance : la présence virtuelle est un aspect important de la formation médiatisée.

Face à cette nouvelle méthode de travail, une adaptation humaine est aussi nécessaire : l'Ergonomie est la science qui s'occupe des études concernant l'adaptation du travail à l'homme. Elle porte sur l'amélioration des postes et de l'ambiance de travail, sur la diminution de la fatigue physique et nerveuse, sur l'enrichissement des tâches.

Notre objectif dans ce projet est d'étudier et examiner quelques méthodes qui donnent le contrôle, en temps réel, à des machines et à des équipements à partir d'un micro-ordinateur. On va regarder par la suite comment manipuler ces équipements à travers le réseau. L'accès à des machines à l'aide d'un ordinateur n'est pas nouveau mais

le contrôle et la manipulation synchrone à distance représente un défi et un problème à résoudre.

On va commencer tout d'abord par le contrôle d'un moteur à courant continu, puis d'un simulateur des procédés qui permet un contrôle de niveau, de température et de pression. On va utiliser le langage de programmation graphique LabVIEW de National Instruments qui possède des outils intégrés pour l'accès à des équipements et le contrôle à distance.

Dans le premier chapitre, on présente les éléments d'un laboratoire virtuel les définitions et les rôles de chaque élément. On fait aussi une étude générale sur la technologie et les outils nécessaires pour créer cet environnement tels que l'Internet, les réseaux, la technique de communication entre un micro-ordinateur et un instrument physique, ainsi que les langages informatiques appropriés. On fait aussi une étude sur différents aspects du laboratoire, les avantages et les inconvénients et finalement on présente un modèle d'un environnement virtuel.

Dans le deuxième chapitre, on trouve une présentation sur l'identification, la modélisation et la notion de base d'un système dynamique. On présente une étude théorique sur les machines électriques en général, les modèles mathématiques et la régulation. Ensuite on fait une étude sur le principe de fonctionnement d'un moteur à courant continu, la modélisation, la fonction de transfert et les calculs théoriques nécessaires pour l'asservissement ainsi que le réglage des paramètres d'un régulateur PID en boucle ouverte et fermée.

Dans le troisième chapitre, on explique l'accès à distance d'un simulateur de procédés. On y trouve une description de ce dernier et de la méthode utilisée pour avoir accès à distance. On y trouve également une explication sur la communication entre

LabVIEW et l'équipement physique par les ports RS232 et RS485, puis on présente les résultats de l'expérimentation.

En annexe, on trouve une représentation détaillée sur l'outil de travail qui est le logiciel LabVIEW avec ses fonctionnalités et son large domaine d'applications. On trouve aussi une explication sur la conception d'un encodeur et en particulier un encodeur incrémental. On trouve également une présentation de l'utilisation du Code Interface Node (CIN) pour appeler un code externe à partir de LabVIEW.

CHAPITRE 1

REVUE DE LA LITTÉRATURE

Jadis, l'information se présentait sous forme statique (livres) et l'enseignement était "sur place". Mais à cause de la diversité des connaissances, actuellement croissante, l'enseignement traditionnel par une seule institution qui dispose des matériels coûteux et aussi perfectionnés est devenu difficile, et grâce à ce progrès technologique, l'information est devenue dynamique de part sa présentation en hypertextes et sa forme variée (textes, images, sons, vidéos) ; en plus l'accès à une multitude d'informations est devenu possible et les formes en sont multiples : Banques de données informatisées, cédéroms, réseaux informatiques et Internet, etc. Le développement rapide des moyens de télécommunication favorise le partage des ressources entre les institutions de formation et de recherche d'une part et les industries ou employeurs de l'autre : Cette association s'impose d'elle-même à cause du virage anticipé vers la formation continue en milieu de travail.

La littérature des dernières années s'est enrichie avec ce modèle d'enseignement. La faculté de Ceske Budejovice [2] qui s'intéresse à la physique des particules et à la physique des hautes énergies dans un sens large a mis au point un laboratoire de physique virtuel vu les coûts prohibitifs des équipements réels et pour plusieurs autres raisons. Les méthodes utilisées sont proches de la réalité virtuelle.

Dans cette faculté, un groupe de physiciens s'occupe de la constitution d'un environnement complet pour l'enseignement de la physique sous forme d'une simulation d'un laboratoire scientifique. Ce projet a été réalisé pour atteindre les objectifs suivants :

Encourager les jeunes à étudier les sciences de l'ingénieur et les bases sous-jacentes de la physique et pour rendre plus efficace la formation des physiciens.

Ce laboratoire est fondé sur une architecture constituée par un système expert et un système à base de connaissance qui comprend :

- Les documents didactiques constituant la charpente du Laboratoire Virtuel. Toutes les informations (textes, images, graphes, schémas, animation, notes, etc.) sont structurées à l'aide de liens hypertextes
- L'environnement informatique servant à la mesure, à l'acquisition et au traitement des données pour diverses expériences
- La modélisation et la simulation, soit en utilisant des méthodes préexistantes soit en en créant de nouvelles.
- L'étude expérimentale de cas d'écoles, de démonstrations, d'utilisation d'outils d'aide à l'enseignement
- Le matériel audiovisuel
- Les outils de Bureautique : traitement de texte, tableurs, dictionnaires électroniques, etc.

Les méthodes utilisées sont proches de la réalité virtuelle. Les étudiants ayant acquis des connaissances de base doivent également apprendre à travailler indépendamment de façon créative et inventive. La première phase qui est relativement passive, concerne la compréhension en profondeur de la théorie et son application, par la suite sur des problèmes concrets. Les exercices théoriques et les expériences de laboratoire deviennent alors la source d'une pédagogie basée sur l'apprentissage.

Le laboratoire virtuel mis au point a plusieurs avantages comme le fait de faire sentir aux étudiants qu'ils sont dans un vrai laboratoire, similaire à celui d'un institut scientifique. Ils peuvent partir d'une situation modélisée pour observer, créer des

hypothèses, examiner différentes alternatives, évaluer les données mesurées, les confronter avec les hypothèses et écrire un article en vue d'une publication. Les avantages sont aussi la sécurité, l'économie de temps, d'espace, de coûts, de personnel, l'accès rapide et les aspects didactiques. Le fait aussi que ce laboratoire soit accessible par le réseau permet aux étudiants de communiquer avec leurs professeurs, des spécialistes et des collègues d'autres institutions scientifiques à travers le monde. L'accès aux bases de données de nombreuses bibliothèques est également possible par ce moyen.

Les inconvénients de ce Laboratoire de physique Virtuel sont : la complexité et la couverture de diverses activités. Il est difficile d'atteindre le niveau de qualité souhaité. Il faut aussi étudier le cas de la langue à utiliser anglais ou autre, le cas d'adapter l'environnement à des cultures multiples, de faire joindre les autres départements à cette entreprise comme ceux de la pédagogie, de l'histoire, de la musique, de la psychologie, etc.

Un deuxième modèle est celui du laboratoire virtuel français [3]. En France, on s'intéresse aussi aux canaux du réseau mondial informatique. On s'intéresse aux dernières technologies, mais on ne s'interroge pas sur l'évolution de l'utilisation de l'Internet. Tout le monde dit que la Société de l'Information, c'est l'avenir ! Mais personne ne fait des recherches dans ce domaine.

Plusieurs questions se posent lors de ce débat, le principal souci est de savoir où va l'Internet et quelles doivent être les fondations de cette société de l'information.

Le chapitre de l'Internet Society (Isoc.) plaide pour la création d'un "laboratoire virtuel français".

Des chercheurs de tout horizons pourraient y confronter leurs analyses, leurs études prospectives. L'idée n'est pas de créer une nième entité au sein du CNRS, (le

Centre National de la Recherche Scientifique) ou de tout autre organisme scientifique. Une telle entité manquerait de réactivité, au regard de la fulgurante évolution des nouvelles techniques de communication. Il s'agit au contraire de provoquer des échanges entre chercheurs par le biais d'un forum de discussion pour commencer, afin de tenter de percevoir, dans des délais raisonnables, les nombreux changements qui s'opèrent dans ce secteur. Les scientifiques présents à Autrans semblent convaincus de l'utilité d'un tel outil de veille technologique. Il reste à formaliser sa création, le plus rapidement possible "Il ne faut plus sous-estimer l'ampleur des mutations économiques et sociales que provoque l'Internet" : Le leitmotiv de l'ISOC France (Internet Society France) sonne comme un avertissement.

Un autre modèle de laboratoire virtuel est celui de Lawrence Berkeley National Laboratory (LBNL) [4]. Ce Laboratoire Virtuel, près de San Francisco, permet de manipuler à travers Internet des équipements scientifiques lourds pour la réalisation d'expériences à distance. Cette première grandeur nature en matière de télé science a été réalisée par le Centre National de Microscopie Électronique du LBNL : à partir de leur laboratoire à 3000 km de Berkeley, des chercheurs ont pris le contrôle, à travers leur micro-ordinateur relié à Internet, du microscope électronique le plus puissant des États-Unis (une machine haute de 3 étages!) pour étudier en temps réel l'évolution d'un échantillon d'alliage avancé soumis à des contraintes thermiques

Pour plus d'une décennie, les scientifiques étaient obligés de voyager au laboratoire national à Ernest Orlando Lawrence Berkeley pour travailler sur un microscope électronique à haute tension. Ce laboratoire a pris les mesures nécessaires pour faciliter l'utilisation de ce microscope à distance.

La principale difficulté avec le microscope réside dans la nécessité de mettre au point un système de visualisation sur un échantillon qui se déforme à l'intérieur d'un caisson d'isolation, et c'est en dépit du décalage temporel inhérent à l'utilisation du

réseau Internet. En fait, les scientifiques du LBNL ont résolu cette difficulté en automatisant la mise au point par un système performant d'algorithmes regroupés sous le terme générique de "vision par ordinateur". C'est l'ensemble de ce système qui peut désormais être piloté à distance.

Le laboratoire a créé des outils interactifs pour le calcul en ligne, qui permettront aux scientifiques de manipuler l'instrument, faire des expériences et observer des images à partir de leurs propres bureaux à travers l'Internet.

L'observation de changements sur un échantillon dans le microscope n'avait jamais été faite à distance. Une démonstration par des scientifiques du Laboratoire de Berkeley a eu lieu récemment et a causé une sensation mineure. Pendant la réunion internationale annuelle de Microscopy Society of America à Kansas, une démonstration de répétition a été exigée par les scientifiques qui avaient manqué le premier. De 3000 Km loin de Berkeley, les chercheurs ont employé un ordinateur pour commander le microscope, ils ont chauffé un spécimen d'alliage avancé et ils ont observé la progression subséquente de changements structuels sur le moniteur d'ordinateur.

La démonstration annonce un nouveau phénomène, l'apparition en ligne "des laboratoires virtuels". Comme l'ordinateur et la technologie de réseau progressent, des ressources scientifiques principales comme le microscope Berkeley deviendront accessibles de tout le monde. Les collaborateurs aux emplacements différents conduiront une expérience et discuteront des résultats tandis qu'il bat son plein.

La télécommande des machines n'est pas nouvelle, mais l'opération éloignée d'un microscope électronique a cependant accompli quelque chose presque impossible. C'est la difficulté de gérer les retards légers du temps mais inévitables, qui arrivent avec l'utilisation d'Internet.

Le microscope est équipé d'une chambre expérimentale pour soumettre un échantillon à une gamme de conditions différentes comme le chauffage, le refroidissement, la tension extensible et le changement des pressions atmosphériques. Au niveau atomique et moléculaire, les choses se déplacent vite.

Un opérateur doit être assis à côté du microscope pour ajuster des cadrans et pour tenir le secteur du spécimen à étudier dans le champ de vision. Imaginez de le réaliser à travers Internet, qui a un retard de temps analogue au retard de voix qui arrive pendant un appel téléphonique international. Les commandes de l'opérateur éloigné du microscope arrivent en retard.

Pour éviter cette limitation de réseaux d'ordinateur, les chercheurs automatisent sur place la position et la concentration du microscope. Cela est rendu possible par le développement d'algorithmes avancés de vision par ordinateur.

Les expérimentateurs à des emplacements éloignés seront capables de manipuler le microscope, changer l'agrandissement, parcourir l'échantillon et changer son orientation et déclencher une gamme de conditions expérimentales. Les collaborateurs le feront par un environnement de calcul qui inclut les outils de communication vidéo nécessaires.

On trouve également plusieurs articles qui traitent du même sujet tel que le télé laboratoire et la formation à distance, parmi ces articles il y a celui de l'université de Hagen en Allemagne [5]. Dans cet article on parle du travail collaboratif entre un groupe des étudiants en utilisant le télé laboratoire, les étudiants ont accès à ce laboratoire et peuvent faire des expériences, en groupe, à travers le réseau d'Internet. L'environnement crée donne aux étudiants la possibilité de discuter les résultats de leur travail comme il permet l'interaction entre eux. L'architecture de cet environnement est basée essentiellement sur des applications client serveur qui sont écrites avec le langage de

programmation Java. Avec cette nouvelle technique, les étudiants ne sont plus obligés de se présenter physiquement dans les laboratoires. En plus plusieurs matériels et ressources sont disponibles à distances et des équipements uniques et coûteux peuvent être partagés entre plusieurs universités. Trois universités allemandes participent à ce projet dont le but est la mise en œuvre d'un environnement expérimental basé sur le réseau. Les étudiants peuvent être éloignés géographiquement mais connectés localement pour effectuer leurs expériences. Cet article explique la technique et les outils nécessaires pour construire ce laboratoire et présente les résultats préliminaires de ce projet.

Un autre projet d'un laboratoire virtuel est celui de l'UCF (University of Central Florida Source) [6]. L'objectif de ce projet est de développer un laboratoire virtuel qui permet aux professeurs et aux étudiants d'accéder à des instruments physiques, de faire des conférences vidéo, de configurer et de contrôler des équipements, en temps réel et en utilisant l'Internet.

CHAPITRE 2

LABORATOIRE VIRTUEL : FORMATION ET ASSISTANCE TECHNIQUE MÉDIATISÉE

2.1 Introduction

Dans ce chapitre nous présentons le laboratoire virtuel, son objectif ainsi que ses éléments avec leurs définitions et le rôle de chaque élément dans l'ensemble. On va citer aussi des outils et des technologies qui peuvent être utilisés pour construire un tel laboratoire ou environnement. Nous parlerons à la fin des avantages et des inconvénients.

2.2 Buts

Un laboratoire virtuel a plusieurs objectifs autant éducatifs que techniques. Son but essentiel est de permettre l'expérimentation indépendamment de la distance, d'une façon similaire à celle réalisable dans les laboratoires conventionnels. Avec les moyens technologiques présents on espère atteindre les mêmes objectifs que ceux d'un laboratoire conventionnel et même d'une façon plus efficace pour certains : c'est le télé-présentiel.

D'une manière spécifique, le laboratoire virtuel vise d'après sa formation à créer un nouveau système d'apprentissage qui sert à construire les connaissances chez les apprenants à travers l'accès à distance à un ensemble de matériels et/ou à des logiciels simulant le fonctionnement de dispositifs. Cette nouvelle méthode met en relief la nécessité de travail en groupe et d'échange d'information notamment entre enseignants-

utilisateurs et utilisateurs-utilisateurs afin de transformer l'apprentissage en un travail constructif guidé par la simulation et l'exploration. Bien que le processus d'apprentissage soit de loin amélioré, les mêmes stratégies pédagogiques sont conservées mais présentées dans le cadre de la technologie actuelle et dotées de toutes ses performances : Elle permet au public l'accès à des bases de données expérimentales afin de découvrir des lois qui sont expliquées au travers des classes théoriques et d'acquérir par la pratique des procédures spécifiques de sa spécialisation comme apprenant.

Cette méthode incorpore d'une manière indirecte plusieurs structurations et plusieurs buts liés aux connaissances et aux attitudes qui cherchent d'une manière ou d'autre à reformer l'étudiant. La façon, dont l'expérience est présentée, favorise la pratique réflexive chez l'étudiant : ses connaissances sont reconstruites par son auto-évaluation et par sa propre découverte de ses points faibles. C'est dans cette formation basée sur le travail individuel que réside l'intelligence du système : l'apprenant par la reconnaissance de ses forces découvre ses dons et augmente son estime de soi.

Les objectifs ne sont pas restreints au domaine éducatif mais aussi à la technologie. L'exploration des infrastructures des réseaux de télécommunication adaptées aux diverses situations éducatives permettra d'élargir les champs d'application et de les rendre à la portée d'un public illimité.

D'une manière plus générale, le but du projet est de réorganiser et de coordonner, d'une façon adaptée à l'évolution de la technologie, les activités synchrones ou asynchrones tels qu'elles se déroulent dans les laboratoires traditionnels de recherche et/ou d'enseignement. On ouvre les horizons à diverses formes de stratégies pédagogiques telles que les jeux, la simulation, l'exploration guidée, le compagnonnage, le parrainage, l'apprentissage par adaptation et les conseils. On conçoit expérimentalement une interface de télécommunication destinée à supporter dans un 1^{er}

temps les expériences du laboratoire pour enfin s'étendre à tous les domaines. Les bases de données des mesures expérimentales sur le réseau sont créées et utilisées d'une manière interactive ; la bande passante des réseaux et le protocole de transmission sont aussi développés afin de répondre aux diverses situations.

Pour pouvoir aborder ce thème, plusieurs notions doivent être mises en relief [7].

2.3 Définitions

Tout d'abord, commençons par éclaircir l'idée de laboratoire virtuel : Qu'entend-nous par " laboratoire virtuel?? ". Qui sont les apprenants, les formateurs...? Quel est le rôle de chacun d'eux ? Et quel lien les relie ?

2.3.1 Laboratoire

Un "Laboratoire" est défini comme étant un local spécialement équipé par les appareils et les produits destinés à des recherches scientifiques ou techniques, à des analyses biologiques, à des travaux photographiques, à des travaux de reproduction et à des essais individuels.

2.3.2 Virtuel

Tandis que "virtuel", par opposition à formel, signifie ce qui n'est point proprement et précisément une certaine chose, mais qui en a la force et la vertu. Il n'existe qu'en puissance, sous une forme abstraite.

2.3.3 Laboratoire virtuel

En fait un laboratoire virtuel est un environnement de travail réparti dans le temps et qui est partagé par plusieurs personnes qui collaborent et coopèrent pour effectuer des tâches, n'appartient en fait à aucune d'entre elles, est accessible uniquement sur réseau télématique ou sur Internet. Il donne l'illusion de réalité. En effet, il permet d'effectuer des tâches inhérentes à différentes disciplines scientifiques ou technologiques en tenant compte de certaines barrières telles que l'impossibilité de manipulation physique directe et les limitations des possibilités d'observation sensorielle (vue, toucher, ouïe).

En d'autres termes, afin d'être un environnement électronique disponible et facilement accessible par chaque utilisateur pendant l'exécution de sa tâche, le laboratoire virtuel rassemble sur un même support des moyens audiovisuels (textes, sons, images fixes et animées) et des moyens informatiques (programme, données) pour les diffuser simultanément et de manière interactive.

2.3.4 Apprenant

Apprenants ou étudiants sont des termes utilisés pour désigner la personne qui est en cours du processus d'apprentissage.

L'apprenant acquiert et traite les informations reçues à travers l'écran de son ordinateur afin de réaliser l'expérience désirée et de présenter un travail écrit. Dans ce cadre, beaucoup d'outils lui sont fournis. Il dispose de diverses ressources technologiques, d'une base de données requise pour la réalisation de ses activités d'apprentissage, des notes explicatives, du lexique et des liens Internet : Il a accès à

plusieurs interfaces permettant la communication entre apprenants d'une part et entre apprenants et formateur d'autre part. Un coin de travail pour la réalisation des activités du projet est à sa disposition afin d'illustrer son sujet par des images, des sons, voir des séquences vidéo (travail multimédia). Il bénéficie d'un forum de discussion où il peut s'exprimer et il peut suivre le déroulement de son cours dans la section des activités d'apprentissage.

2.3.5 Formateur

Le formateur est celui qui fait la mise en œuvre de séances interactives d'instruction de travaux pratiques à l'aide des scénarios appropriés. Son rôle est pédagogique, social et technique.

Il a accès aux scénarios d'apprentissage. Il dispose d'une section lui permettant d'assurer le suivi des apprenants, de les évaluer et de toutes les autres ressources du système qui lui assurent le soutien au développement des projets en question. Les formateurs deviennent des animateurs et des facilitateurs.

Les apprenants et les formateurs effectuent des tâches respectives, séparément tout en restant en contact selon divers moyens.

2.3.6 Administrateur ou gestionnaire

L'administrateur doit avoir la possibilité de contrôler et de gérer l'ensemble des contenus présentés avec une très grande liberté tant dans la présentation que dans la structure. Il assure la mise à jour du système ainsi que la disponibilité et l'accessibilité des équipements.

2.4 Potentiel (en dimension culturelle)

Avec ce concept, de nouveaux horizons sont élargis et de nouvelles façons d'apprendre, d'enseigner et de travailler sont incorporés. Ce qui représente la puissance de ce système de télé apprentissage. Une nouvelle école sans frontière est née afin de répondre aux besoins d'élèves d'origines et de cultures diverses : Les institutions de formation deviennent une aire ouverte au monde puisqu'elles forment des ingénieurs ou des techniciens capables d'apprendre par eux-mêmes tout au long de leur carrière. La barrière entre le travail et la formation, en plus les contraintes d'espaces et de temps sont franchies.

Le potentiel de ce concept est indépendant de l'espace et du temps. Les étudiants quant à eux peuvent travailler à temps partiel en parallèle ou effectuer des stages à l'extérieur de leur ville. Le professeur ou l'assistant peut prendre le temps de consulter ses ressources avant de donner une réponse concise ou en parallèle assister à des enseignements ailleurs.

Le développement des ressources télématiques sera accéléré à force d'être explorées. Beaucoup de moyens surtout de communication seront mis en œuvre pour donner immédiatement les rétroactions et minimiser les délais.

De même en plus de supprimer les frontières des distances, ce concept permet de diminuer la différence entre les pays : certains ne seraient plus considérés comme des consommateurs ou des fournisseurs de matières premières mais plutôt comme des collaborateurs dans le domaine des connaissances qui représentent maintenant la base de tout développement. Un modèle d'échange actif entre institutions nationales et institutions étrangères est établi.

2.5 Aspects pédagogiques

Le laboratoire virtuel est un système qui crée un environnement d'apprentissage à partir d'outils informatiques, logiciels et réseaux, permettant à chaque participant de jouer son rôle dans les activités d'apprentissage.

L'étude concernant le laboratoire virtuel nécessite une étude diversifiée sur le thème des nouveaux médias d'apprentissage (processus d'apprentissage, d'acquisition de connaissances, et d'expérimentation) et sur les outils disponibles (environnement pédagogique informatisé, les stratégies et les techniques pédagogiques ouvertes) puisque l'ambiance et la démarche assurées sont celles d'un laboratoire réel. Ceci se traduit par la formation de toute une génération à la nouvelle conception d'apprentissage à distance. La bonne connaissance du potentiel éducatif du public (dans un sens ou un autre) et le critère de but définissent la méthodologie.

Plusieurs facteurs, dépendant de l'infrastructure adoptée et des ressources informatiques, mettent en œuvre un système de formation à distance ouvert qui favorise une dynamique évolutive du groupe en situation d'apprentissage.

Le premier pas vers ce monde de télé apprentissage est d'approfondir les concepts, les lois et les phénomènes scientifiques (ou technologiques) relatifs à l'expérience en question ceci afin de la traduire en une séquence animée. L'ensemble de ces notions est stocké dans une base de données à architecture ouverte permettant un affichage varié des différentes techniques de communication (groupes de discussion, résultats de recherche, articles...). Ainsi une seule interface permet au site de servir aussi bien de site d'actualités que de site de référence ou encore d'un lien de discussion pour les acteurs du domaine.

L'apprenant pour sa part doit être formé à cette nouvelle approche et doit avoir des habiletés techniques et scientifiques assez bien adaptées aux démarches scientifiques et au travail relatif au laboratoire. Au cours de son apprentissage, il dispose de plusieurs outils de communication et de plusieurs interfaces (une grille ou un cahier de charges) lui permettant de communiquer avec tous les participants et d'enregistrer ses commentaires au cours des simulations. Il doit aussi avoir les moyens de mettre son expertise et ses ressources techniques à la disposition des autres usagers. Il a accès direct ou indirect à ces diverses ressources d'apprentissage : ces dossiers, ainsi créés pendant les sessions, sont retenus en fonction de leurs objectifs et permettent ainsi de reconstituer les profils d'apprentissage individuels et ceux des groupes. En fait ces évaluations et ces remarques suggérées permettent l'amélioration de la performance des activités d'apprentissage.

L'apprenant dispose de même de toutes ces bases de données. Son rôle n'est pas limité à recevoir et assimiler un contenu et ensuite le transmettre mais une stratégie pédagogique évolutive assure un lien de communication entre tous les assistants de l'activité considérée.

Les sites construits dynamiquement à partir de données résidentes sur le serveur en constante évolution et systématiquement mises à jour lors de la consultation.

Ce concept repose essentiellement sur les technologies appliquées aux transducteurs (capteurs et actionneurs) et sur les techniques d'acquisition des données (conversion, traitement, conditionnement et transmission des signaux).

2.6 Aspects techniques

Pour la réalisation d'un laboratoire virtuel, la nécessité de plusieurs systèmes de communication est nécessaire afin d'offrir des mécanismes de transfert entre ordinateur - machine d'une part et ordinateur-humain d'autre part et assurer l'analyse et le besoin des échanges entre tous les utilisateurs.

Un progiciel doit être mis en œuvre et doit permettre d'une part de commander par l'entremise d'une interface des dispositifs et d'autre part la réalisation en temps réel d'une certaine expérience. Le logiciel et la plate forme assurent la rapidité de calcul. Tous les constituants de l'interface PC-humain seront des éléments hypertextes, hypermédias et des ressources Internet qui permettent de mettre en ligne des simulations, des cours interactifs et des programmes complets de formation à distance et de les gérer sans aucune difficulté. Ceci assure la flexibilité des outils tandis que l'interface machine – PC est responsable de la numérisation et de l'acquisition de données.

Il faut penser aussi à créer un système de projection stéréoscopique pour grandes audiences afin de donner à chaque étudiant la chance de participer aux activités d'apprentissage. Le tout intégrant les notions de réalité virtuelle permettant une interaction en temps réel. Le professeur pourra, à l'aide des outils, guider l'audience dans un monde tridimensionnel interactif.

Cet environnement interactif doit assurer la communication en temps réel et doit être caractérisé par :

- La puissance de calcul élevée.
- La mémoire à grande capacité et à accès rapide.
- Les bases de données à grande capacité, locales, distantes ou distribuées.
- Les réseaux à haute vitesse et à grande capacité.

2.7 Outils et approches

Plusieurs technologies et outils peuvent être utilisés pour construire un laboratoire virtuel. En effet ces outils doivent être à la fine pointe de la technologie disponible. L'environnement de télé-laboratoire doit tenir compte de trois composants, comme l'illustre la figure 2.1 :

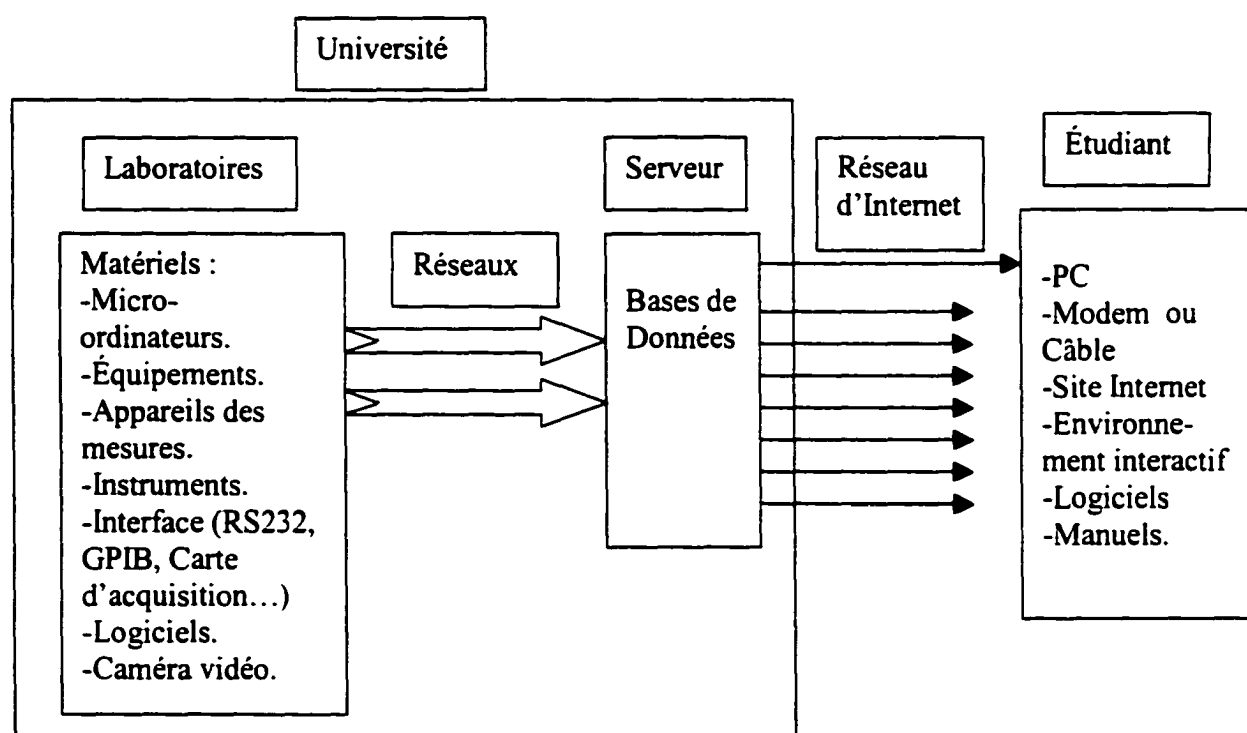


Figure 1. 1 : Composants du laboratoire virtuel.

- 1- Côté étudiant (interfaces, logiciels, site Internet, PC, modem, environnement interactif, documents et guides, etc.).
- 2- Côté Réseau ou liaison (Type de réseau, protocole de communication, Internet, outils de communication : Vidéo, chat, conférence, etc.).

- 3- Côté laboratoire ou Université (Matériel de communication : RS232, GPIB..., équipements, logiciels, etc.).

Nous allons, dans ce qui suit, détailler chacun des éléments mentionnés plus haut :

1- Côté étudiant

Dans cette direction, il faut développer les interfaces et l'environnement. Chaque essai ou chaque expérience a sa propre interface, sans oublier l'aspect pédagogique parce que le but avant tout c'est fournir l'information.

L'étudiant aura accès à toutes les ressources à travers un site web efficace; ce site se comporte comme un environnement virtuel et non comme un site web traditionnel, tout en offrant des informations mises à jour en temps réel par les administrateurs. Il permet aux étudiants la manipulation, le téléchargement des manuels d'utilisation et des guides pour avoir toutes les informations nécessaires pour pouvoir utiliser les interfaces et les outils mis à ses dispositions; il donne aussi un accès à des équipements et des matériels pour réaliser ces travaux à distance. C'est cette constante corrélation des informations qui fait la force de ce site.

Chaque étudiant doit aussi pouvoir communiquer avec l'enseignant et avec ses collègues. Il peut exprimer, poser des questions et s'informer quand il a besoin. L'utilisation d'un courrier électronique peut être un moyen de communication.

N'importe quelle plate-forme peut être utilisée; elle doit être compatible avec l'environnement créé. Les étudiants auront alors plus facilement accès à ces nouveaux moyens d'apprentissage. Il leur sera possible de réaliser des séances de laboratoire de leur lieu d'habitation, aux heures qui leur conviennent le plus. La liaison peut se faire à

l'aide d'un modem ou de l'Internet à haute vitesse. Le langage informatique utilisé doit permettre la télémessure et la télécommande sur réseaux.

2- Côté réseaux ou liaison

Les nouvelles technologies d'information et de communication ont rendu plus accessibles la communication entre les personnes. La communication à distance est devenue possible à moindres coûts et de manière presque instantanée grâce au courrier électronique, chat, vidéoconférence. La distance est réduite et l'information est devenue pleinement accessible, interactive et attrayante.

Depuis quelques années, un peu partout dans le monde, la demande de débits élevés exigés par de nouveaux services ne cessent de croître. L'apparition des applications multimédias, l'évolution vers les applications client-serveur et l'augmentation de la puissance des postes de travail génèrent un besoin d'infrastructure souple et de haute performance, pour les réseaux publics et privés.

Mais la vitesse de transfert des données demeure un problème important pour le bon fonctionnement des systèmes d'information, plusieurs techniques sont actuellement utilisées pour contrôler la bande passante requise : compression des données transmises, serveurs d'applications, réplication des données transmises...

La transmission de flot de données multimédia sur Internet est maintenant une réalité qui est déjà implantée dans nos ordinateurs; la réception de ces flots avec un niveau de qualité acceptable à la destination représente un défi pour les réseaux. La recherche sur les réseaux à haut débit porte essentiellement sur la technologie et les applications. Il faut développer des applications éducatives génériques par rapport à la topologie des réseaux à large bande qui peuvent être adaptés à diverses situations.

Il faut penser à créer un réseau à grand débit dans le but de permettre aux usagers d'accéder à des ressources et à des données à distance. Pour y répondre, les recherches doivent entreprendre le développement de réseaux à large bande, ayant des vitesses de transmission pouvant dépasser le G.Bit/s. Un tel réseau permet de faire face à l'accroissement des trafics multimédias et des données. Il permet aussi d'améliorer l'utilisation des ressources du réseau. Le partage des ressources est un problème à résoudre pour la gestion et le contrôle dans le but d'optimiser le réseau et satisfaire au mieux l'utilisateur.

3- Côté laboratoire

Dans cette partie, on doit faire des recherches pour définir les solutions techniques possibles, rassembler les outils nécessaires pour construire cet environnement (Cartes d'acquisition, ports séries, ...) et on doit voir aussi toutes les possibilités offertes par ce type d'outils au niveau pédagogique. L'objectif sera donc l'utilisation des technologies modernes de l'information pour l'analyse, la conception et la réalisation d'un environnement informatique multimédia capable de supporter des activités d'apprentissage à distance, d'une manière coopérative. Les évolutions dans ces domaines demandent des efforts supplémentaires et le développement doit se faire sur les outils qui peuvent assurer la représentation et la manipulation des concepts des sciences, en particulier liés à l'expérimentation et aux travaux pratiques.

Les travaux initiaux consisteront essentiellement en la réalisation d'un environnement permettant la visualisation interactive de mondes virtuels ayant des objectifs scientifiques. L'environnement informatique servant à la mesure, à l'acquisition et au traitement des données pour diverses expériences. La figure 2.2 nous montre un modèle de cet environnement.

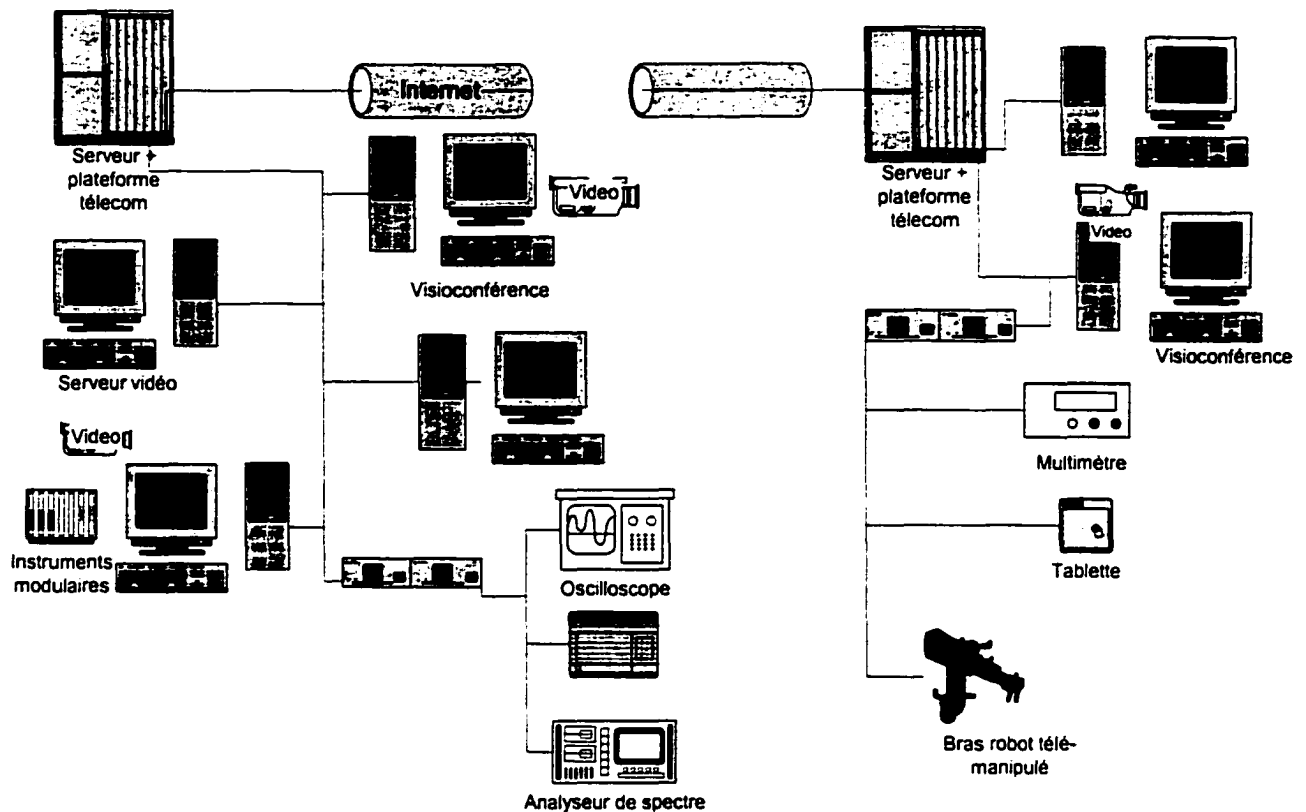


Figure 2. 2 : Un modèle d'un environnement virtuel.

Le but est de faire en sorte que les étudiants se sentent dans cet environnement virtuel comme s'ils étaient dans un laboratoire réel. Ils peuvent observer, créer des hypothèses et examiner différentes alternatives, évaluer les données mesurées et les confronter avec les hypothèses.

Il faut aussi développer le processus d'enseignement associé et il est nécessaire d'évaluer si cette approche ou cette nouvelle méthode peut permettre d'atteindre le but pour lequel le laboratoire a été construit. Il est aussi important de vérifier que l'acquisition des connaissances par l'apprenant est assurée.

On va parler de quelques outils qui représentent des éléments indispensables pour construire ce laboratoire, avec leurs définitions et leurs caractéristiques :

2.7.1 L'Internet

L'Internet facilite le partage de l'information entre les gens, permet l'accès à des bases de données rendues accessibles à tous. Il n'y alors plus de barrières à l'information, il est entré dans les mentalités des gens et il représente un espace public à l'échelle de l'humanité. Il permet aussi une égalité des chances pour l'ensemble des étudiants. Les services qu'ils peuvent obtenir grâce à l'Internet sont de plus en plus diversifiés et ne touchent plus uniquement, comme cela était le cas au début, les services de documentation.

Le site informatique doit être très efficace mais pas très grand afin qu'il ne soit pas lourd pour l'utilisateur qui le consulte et il doit correspondre à une demande stratégique. Toutes les informations (textes, images, graphes, schémas, animations, notes, etc.) sont structurées à l'aide de liens hypertextes. Ce site doit avoir une grande capacité d'écoute vis à vis de l'utilisateur. On peut ajouter des procédés vidéos pour permettre un enrichissement de la communication.

L'ensemble des contenus est stocké dans une base de données, dont l'architecture ouverte permet un affichage varié des différentes sections possibles. La mise en place d'un outil administratif puissant et convivial s'est imposée pour permettre aux administrateurs de contrôler l'ensemble des contenus présentés avec une très grande liberté tant dans la présentation que dans la structure.

L'ICANN (Internet Corporation for Assigned Names and Numbers) a annoncé la création de sept nouveaux domaines Internet qui viennent s'ajouter aux trois domaines génériques, soit les .com, .net et .org. [8]. Désormais, les pages de nos sites web se

déclineront également et notamment en .biz, .tv, .pro ou .museum. Certes, cette réforme va donner un peu d'air au système. Il devenait difficile de trouver un nom de domaine libre. Ainsi, plus aucun nom à trois caractères en .com, .net et .org n'était disponible. Toutes les combinaisons ayant déjà été attribuées.

2.7.2 Le débit des réseaux

Avant toute chose, nous souhaitons faire une rapide analyse des besoins informatiques actuels, quant au volume de l'information transportée. En effet, un utilisateur qui manipule un fichier texte, n'a pas les mêmes besoins qu'un graphiste lorsqu'il s'agit d'échanger des données. De même, nous n'avons pas besoin d'un gros débit pour lire quelques pages Internet, par contre si nous souhaitons visionner un film, faire de la visioconférence, le débit nécessaire se voit décupler ! Voyons avec des exemples :

- Une page de texte prendra par exemple 1200 octets
- Une page de tableur prendra 12000 octets
- Une photographie en format A4 demande 500000 octets

Considérons que nous chargions la page texte pour la lire en 30 secondes, nous avons donc besoin de 1200 octets toutes les 30 secs., soit 40 octets/sec (Vitesse : 320 bit/s). Dans le cas d'une image de bonne qualité le débit varie entre 40 Kilo octets/30s (10 k.bit/s) et 500 Kilo octets/3s (1300 k.bit/s).

2.7.3 L'acquisition de données

Aujourd'hui, la plupart des scientifiques et des ingénieurs utilisent des ordinateurs dotés des technologies PCI, PCMCIA, ISA USB ou ayant des ports

parallèles ou séries pour l'acquisition de données pour effectuer des mesures et des tests ainsi que pour leurs travaux de recherches et études dans les laboratoires. Plusieurs applications emploient des cartes internes (plug in boards) pour acquérir des données et les transférer directement à la mémoire. D'autres utilisent des cartes externes branchées via le port parallèle ou série. Les résultats obtenus dépendent de chacun des éléments suivants :

- 1- Micro-ordinateur.
- 2- Capteurs.
- 3- Matériels pour l'acquisition de données.
- 4- Logiciels utilisés.

2.8 La carte d'acquisition

2.8.1 Généralisation

Les cartes d'acquisition de données aident à acheminer l'information de mesure représentée par des signaux numériques et analogiques, comme nous montre la figure 2.3. Les signaux numériques peuvent venir d'une variété de sources telles que des fermetures de commutateur, des contacts de relais ou des interfaces compatibles TTL. Avec l'interface appropriée ces signaux peuvent être directement lus et traités par des ordinateurs.

Les signaux analogiques viennent des instruments ou des capteurs. Ces signaux ne peuvent pas être directement lus ou traités par des ordinateurs. Ils doivent d'abord être convertis en signaux numériques. Ce processus s'appelle conversion analogique numérique ou CA/N.

Le processus inverse de conversion numérique analogique ou CN/A, change les données numériques pour une tension analogique ou signaux de courants. Plusieurs carte d'acquisition ont des convertisseurs A/N et N/A. Ceci permet la mesure et la commande automatisée des processus et des expériences industrielles de laboratoire.

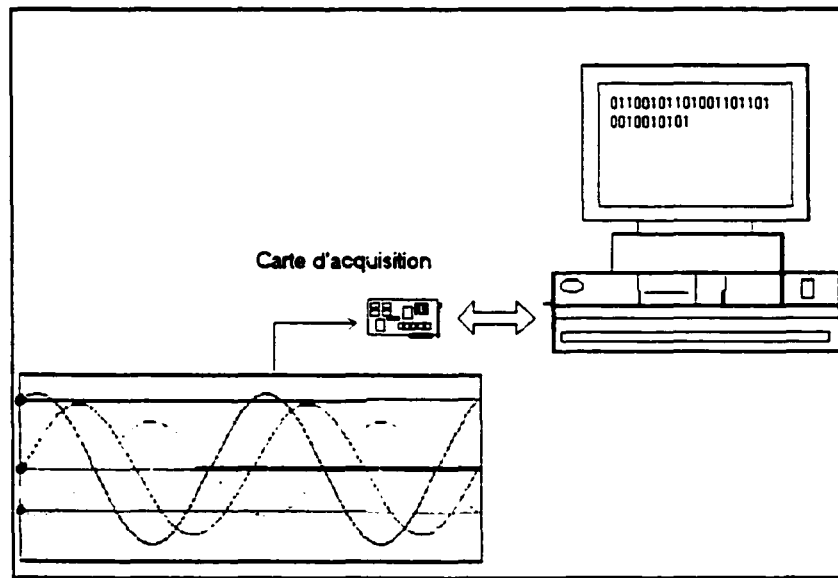


Figure 2. 3 : Une carte d'acquisition pour lire des signaux.

2.8.2 Canaux différentiels vs Asymétriques

Le nombre de canaux d'entrée détermine le nombre de dispositifs qui peuvent être reliés à une interface. Les canaux d'entrée peuvent être asymétriques ou différentiels.

Une entrée asymétrique mesure la tension entre le canal d'entrée et la prise de terre du convertisseur A/N. Chaque canal d'entrée peut être utilisé pour se connecter par interface à un dispositif différent. Le dispositif à mesurer doit fournir un signal qui peut être mis en référence à la masse de l'interface.

Une entrée différentielle mesure la tension entre deux lignes entrées. Ceci exige deux entrées de canaux par dispositif et il a un avantage important par rapport à des entrées asymétriques. Les entrées différentielles peuvent mesurer les dispositifs qui ne peuvent pas être mis en référence à la masse du convertisseur A/N. Les entrées différentielles peuvent également annuler le bruit ou l'interférence des moteurs, des tensions AC ou d'autres sources électriques ou mécaniques qui injectent le bruit dans le capteur ou le câblage au convertisseur A/N.

2.8.3 Convertisseur A/N

Le convertisseur A/N convertit le signal d'entrée analogique en valeur numérique. L'exactitude de la conversion dépend de la résolution et des linéarités du convertisseur. Les erreurs de gain et de l'amplificateur d'entrée affectent également la précision.

La résolution est le nombre de niveaux employés pour représenter l'intervalle d'entrée analogique. Rectifier quatorze convertisseurs de bit, par exemple, peut supposer que 214 niveaux différents et ainsi diviser leur intervalle d'entrée en 16.384 morceaux. Plus de bits rapportent une résolution potentielle exceptionnellement plus élevée. Le gain d'entrée ou l'amplification, peut augmenter la résolution apparente des signaux qui ont une amplitude maximum moins que l'intervalle d'entrée du convertisseur A/N.

La linéarité est une mesure de la façon dont les niveaux sont écartés dans l'intervalle d'entrée. La non-linéarité différentielle c'est l'erreur entre les niveaux adjacents. Si 1 LSB (Least significant bit), bit de poids faible, est plus grand que certains des niveaux numériques non utilisés, alors la résolution réelle du convertisseur est moins que la largeur numérisée de valeur. La non-linéarité intégrale donne la différence entre le niveau numérisé et le niveau idéal. Un convertisseur idéal aurait une non-linéarité intégrale de 1/2 LSB.

2.8.4 Débit

Trois éléments indiquent le débit de A/N : temps de conversion, temps d'acquisition, et temps de transfert.

Le temps de conversion est le temps requis par le convertisseur A/N pour produire une valeur numérique qui correspond à l'entrée analogique.

Le temps d'acquisition correspond au temps requis par les circuits analogiques associés pour saisir un signal.

Le temps de transfert correspond au temps nécessaire pour transférer des données de l'interface à la mémoire d'ordinateurs. Le débit est le taux auquel chacun des ces trois éléments peut être terminé.

Le débit est souvent le facteur le plus important en choisissant une carte d'acquisition de données. Le Théorème de Nyquist indique qu'un signal d'entrée devrait être échantillonné au moins deux fois plus rapidement que la fréquence la plus élevée de ce signal. Ce qui évite le chevauchement. Par exemple pour mesurer exactement un signal 1kHz, le débit minimum du convertisseur A/N est de 2kHz

Les entrées analogiques multi-canal le partagent le même convertisseur A/N. Plus le nombre de signaux d'entrée à convertir est grand, plus le débit de canal est petit. Pour échantillonner 10 canaux à 10 kHz chacun, il faut un système d'acquisition de données avec un débit au moins de 100 kHz.

2.8.5 Amplificateur d'entrée

L'amplificateur d'entrée est utilisé pour amplifier par un gain le signal d'entrée. Le gain de chaque canal d'entrée est normalement placé de telle sorte que l'exploitation des convertisseurs A/N soit au maximum. Quand le gain est élevé, l'intervalle pertinent d'entrée vu par le convertisseur A/N est diminué de sorte que chaque niveau numérisé devient plus fin.

Le bruit et la tension de décalage (DC) de l'amplificateur devraient être également réduits. Le bruit est habituellement référé à l'entrée. Pour trouver ce niveau de bruit réel au convertisseur, il faut multiplier ses spécifications par le gain. La tension peut être indiquée à l'entrée ou être rapportée au convertisseur A/N. Si elle est indiquée à l'entrée, elle doit également être multipliée par le gain. Cette tension est souvent donnée en LSB.

2.8.6 Entrées/sorties numériques

Les entrées/sorties numériques d'une interface d'acquisition de données fournissent les ports bidirectionnels de commande et de mode de niveau TTL qui peuvent être placés et lus par l'ordinateur. Ceux-ci peuvent être employés pour contrôler des dispositifs ou pour contrôler des fermetures de commutateur ou de contact.

2.8.7 Compteurs et temporisateurs

Pour exécuter des conversions multiples dans des intervalles de temps précis, les cartes d'acquisition de données sont équipées de compteurs et de temporisateurs (counter/timers) qui servent à contrôler les conversions de données A/N et N/A. Ils fonctionnent en utilisant un oscillateur interne ou une source extérieure. Cette fréquence

de base détermine la précision des configurations disponibles. Des fréquences plus élevées offrent une meilleure précision.

Quelques cartes fournissent des compteurs et des temporisateurs pour l'utilisateur. Ceux-ci peuvent être extrêmement flexibles et utilisés dans une douzaine de configurations. Quelques applications comprennent la génération d'horloge, les mesures de largeur d'impulsion et de fréquence et la synchronisation externe pour le déclenchement externe compliqué. Elles peuvent être utilisées séparément ou en combinaison avec d'autres compteurs, en utilisant des sources externes ou internes d'horloge.

2.9 Les réseaux

Les réseaux de télécommunication sont ceux qui utilisent la communication numérique. Ils peuvent être classés en fonction de leur étendue. Nous pouvons identifier trois grandes catégories, notamment :

- 1- Les WAN (Wide Area Network).
- 2- Les MAN (Metropolitan Area Network).
- 3- Les LAN (Local Area Network).

Les WAN sont des réseaux étendus généralement publics. Ces types de réseaux permettent les liaisons point à point entre les commutateurs qui constituent sa topologie. Les WAN ont généralement une bonne sécurité.

Le fait que les distances soient importantes, le délai de transmission est un facteur à prendre en considération.

Les MAN, réseaux métropolitains, forment généralement la dorsale (backbone) qui permet à plusieurs réseaux locaux de s'interconnecter. On trouve des MAN dans les campus où la dispersion des bâtiments ou étages ne permettent pas d'avoir un seul réseau local. Le MAN dans ce cas relie les réseaux de chaque département pour en faire un grand réseau.

Les réseaux locaux ou LAN, illustré dans la figure 2.4, ont fait leur apparition dans les années 80 et correspondent à la multiplication des micro-ordinateurs. Comme leur nom l'indique, leur étendue ne dépasse pas quelques kilomètres, leur débit va de plusieurs centaines de K.bits à quelques M.bits. Ils sont faciles à mettre en oeuvre et souvent économiques car les 7 couches de l'OSI *International Standard Organisation* ne sont pas toujours indispensables.

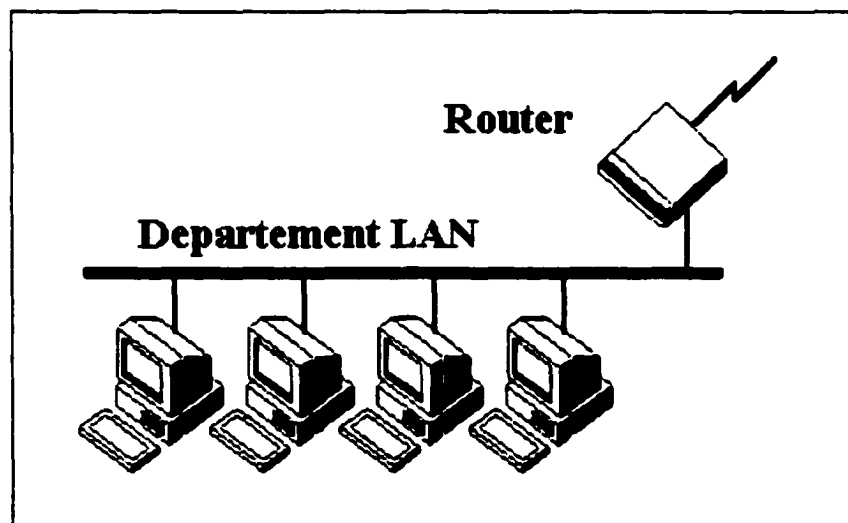


Figure 2. 4 : Réseaux LAN.

Les réseaux locaux font partie du monde des télécommunications, et concernent précisément les besoins d'échanges téléinformatiques à l'intérieur d'une entreprise ou d'un organisme (école, université,...). Ils permettent donc le transfert de données

numériques entre ordinateurs, autorisant les utilisateurs à copier des fichiers ou à partager certaines ressources via le réseau (imprimantes, ...).

2.9.1 Ethernet

Ethernet a été proposé par Xerox Corporation, Palo Alto Research Center (PARC), Digital Equipment Corporation (DEC) et Intel en 1972. Ethernet était techniquement basé sur la spécification d'Aloha. Quelque temps plus tard, ce LAN a été légèrement modifié pour devenir IEEE 802.3.

IEEE 802.3 maintiennent le plus grand marché de tous les protocoles de réseaux locaux. Aujourd'hui, le terme Ethernet est souvent associé à la méthode CSMA/CD (Carrier Sense Multiple Access/Collision Detection) qui est conforme aux spécifications IEEE 802.3. Voir figure 2.5.

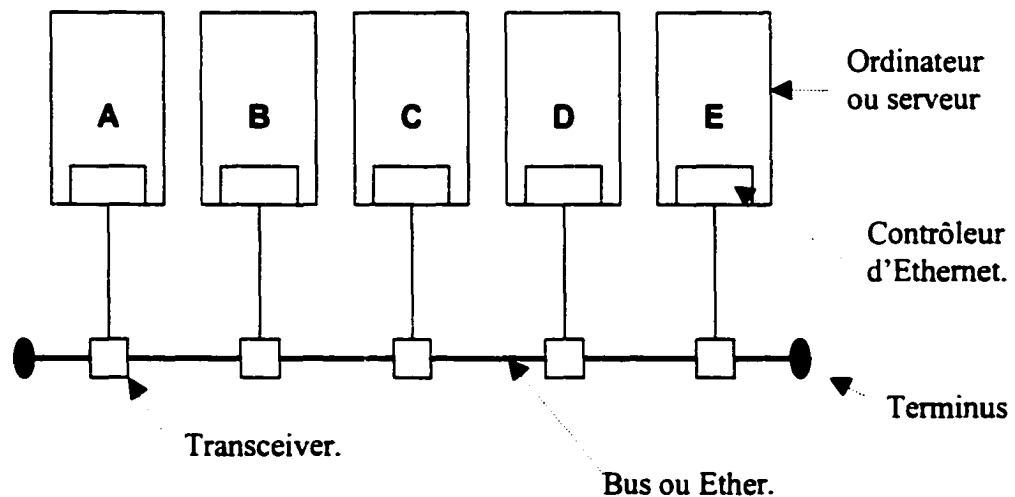


Figure 2. 5 : Ethernet sur LAN (Local Area Network).

L'Ethernet standard a pour caractéristiques :

- 10 M.bps sur un Bus

- Bus composé d'un câble coaxial 10Base-5
- Méthode d'accès au support CSMA/CD

La norme 802.3 impose :

- Une distance maximale entre deux stations de 2500 mètres.
- Un nombre de segment allant jusqu'à 5, soit 4 répéteurs en cascade.
- Un temps d'aller/retour des informations sur le média de 51,2 μ s.
- Une distance minimale entre deux stations de 2,5 mètres, ce qui implique un nombre maximum de nœuds.
- Un nombre donné de stations adressables (1024).

Ethernet a l'avantage d'être un réseau normalisé indépendant de tout constructeur, il est très répandu, aisément modifiable, largement extensible et relativement peu cher. Son seul inconvénient est de supporter assez mal la montée de la charge globale.

2.9.2 VLAN

Abréviation pour LAN *virtuel*, un réseau d'ordinateurs qui se comportent comme s'ils étaient connectés au même fil bien qu'ils peuvent en réalité être physiquement placés sur des segments différents d'un LAN. Les VLANs sont configurés par le logiciel plutôt que le matériel, ce qui les rend extrêmement flexibles. Un des plus grands avantages des VLANs est que quand un ordinateur est physiquement déplacé à un autre segment, il peut rester dans le même VLAN sans aucune reconfiguration du matériel.

2.9.3 VPN

Abréviation pour (*virtual private network*) un réseau qui est construit en utilisant des fils publics pour connecter des nœuds. Par exemple, il y a un certain nombre

de systèmes qui vous permettent de créer des réseaux utilisant l'Internet comme le moyen pour transporter des données. Ces systèmes emploient le codage et d'autres mécanismes de sécurité pour assurer que seulement les utilisateurs autorisés peuvent avoir accès au réseau et que les données ne peuvent pas être interceptées.

2.10 Protocole de transmission

Nous citons ici les deux principaux protocoles de la couche transport d'Internet. TCP, pour *Transmission Control Protocol* et UDP pour *User Datagram Protocol*. Tous les deux utilisent IP *Internet Protocol* comme couche réseau, mais TCP procure une couche de transport fiable, tandis que UDP ne fait que transporter de manière non fiable des datagrammes.

2.10.1 Le protocole UDP

Le protocole UDP utilise le protocole Internet IP pour acheminer d'un nœud à un autre des datagrammes qui lui sont transmis par une application. UDP n'utilise pas d'accusés de réception et ne peut donc pas garantir que les données ont bien été reçues. Ils ne réordonnent pas les messages si ceux-ci n'arrivent pas dans l'ordre dans lequel ils ont été émis et n'assurent pas non plus le contrôle de flux. C'est donc à l'application utilisatrice d'UDP de gérer les problèmes de pertes de messages, duplications et retard.

2.10.2 Le protocole TCP/IP (*Transmission Control Protocol/Internet Protocol*)

Le TCP/IP a été créé en 1974 par le Ministère de la Défense nationale afin d'améliorer et de standardiser le mode de connexion entre machines hétérogènes. En 1978, TCP est fragmenté en TCP/IP, et c'est vers 1980 que le réseau Internet est apparu. À ce moment là le ARPA (*Advanced Research Project Agency*) commença à faire évoluer les ordinateurs en utilisant des protocoles de communication plus élaborés tel

que TCP/IP. Rapidement la quasi-totalité des départements informatiques des universités américaines est connectée. Vers 1989 est créé le WWW (World Wide Web) par Tim Bernes-lee, mais seulement vers le milieu des années 1990 pour le commerce électronique.

L'*Open System Interconnection* est une norme établie par L'*International Standard Organisation*, afin de permettre aux systèmes ouverts (ordinateur, terminal, réseau...) d'échanger des informations avec d'autres équipements hétérogènes. Cette norme est constituée de 7 couches, dont les 4 premières sont dites *basses* et les 3 supérieures dites *hautes*. Le principe est simple, la couche la plus basse (directement au-dessus du support physique) ne peut communiquer directement avec une couche $n+1$, chacune des couches est composée d'éléments matériels et/ou logiciels chargés de « transporter » le message à la couche immédiatement supérieure :

- 7- Application.
- 6- Présentation
- 5- Session
- 4- Transport : TCP
- 3- Réseau : Routeur
- 2- Liaison : Pont
- 1- Physique : Répéteurs

Le TCP/IP utilise IP pour acheminer l'information d'un nœud à un autre, chaque datagramme émis par TCP et encapsulé dans un datagramme IP. Les applications dialoguant au travers de TCP sont considérées l'une comme serveur et l'autre comme client, elles doivent donc établir une connexion avant de pouvoir dialoguer. La fiabilité fournie par TCP consiste à remettre des datagrammes, sans pertes, ni duplication. On utilise pour cela la technique de l'accusé de réception.

Le protocole de contrôle de transmission TCP est responsable de vérifier la livraison correcte de données du client au serveur. Les données peuvent être perdues dans le réseau intermédiaire. TCP ajoute l'appui pour détecter des erreurs ou des données perdues et déclenche la retransmission avant que les données ne soient correctement et complètement reçues.

Le protocole d'Internet IP est responsable de déplacer le paquet de données du nœud au nœud. Ses adresses peuvent être utilisées sur des architectures matérielles différentes (Ethernet, Token-Ring) possédant leurs propres adresses physiques. Il y a nécessité d'établir les correspondances entre adresses IP (logiques) et adresses matérielles (physiques) des nœuds d'un réseau.

Le TCP/IP assignent un numéro(adresse) unique à chaque poste de travail dans le monde. C'est l'adresse IP, une adresse IP identifie d'une manière unique une machine ainsi que le réseau sur lequel elle est située. Chaque adresse est une série de quatre octets dont une partie correspond à l'identificateur du réseau et l'autre partie à l'identificateur de la machine.

2.11 Le protocole GPIB

Le GPIB (General Purpose Interface Bus) fut développé par HP au début des années 70, pour connecter et contrôler des instruments programmables produits par HP et pour simplifier l'interconnexion des instruments de test avec les ordinateurs. Ce nouveau bus devait utiliser un câble standard pour interconnecter de multiples instruments à l'ordinateur. Chaque instrument devait avoir sa propre interface et devait savoir comment répondre à certaines commandes [9].

Cependant, avec l'introduction de commandes numériques et d'équipements de test programmables, il a été nécessaire de créer un standard, le IEEE 488 qui répond à certaines spécifications.

Ce standard a été amélioré par la suite et a conduit successivement à la norme IEEE 488.1 puis au driver IEEE 488.2. Il apporte des atouts supplémentaires au système par la mise en place de commandes de logiciels.

En 1990, fut créé le SCPI (Commandes Standards pour l'Instrumentation Programmable) qui garantit la compatibilité et la configuration de divers instruments et logiciels SCPI avec le GPIB et qui, combiné avec le IEEE 488.2, garantit une meilleure productivité.

Le standard 488.2 a facilité la communication avec un instrument GPIB mais chaque instrument possède encore un ensemble unique de commandes propres à chaque famille d'instruments d'un même constructeur. HP a travaillé sur le problème et a proposé un langage Test et Mesure (TML) qui était basé sur un modèle d'instrument.

Le GPIB est une interface de communication parallèle de 8 bits, avec des taux de transfert pouvant atteindre 8 M.octets par seconde. Il standardise les formats des messages de l'instrument, un ensemble de commandes, une structure de renvoi d'état standard et les protocoles des contrôleurs, ce qui a permis d'unifier le contrôle d'instruments fabriqués par des centaines d'entreprises différentes.

Les contrôleurs ont la possibilité d'envoyer des messages de commande, de communiquer des données sur le bus et de récupérer des données venant des divers périphériques. Il y a un contrôleur dans le système qui est défini comme le contrôleur en charge.

Les périphériques GPIB peuvent envoyer des données, les recevoir et les lire et peuvent même gérer le flux d'informations en envoyant des ordres aux autres périphériques.

Vous trouverez ci-dessous le concept IEEE 488 un contrôleur et des périphériques.

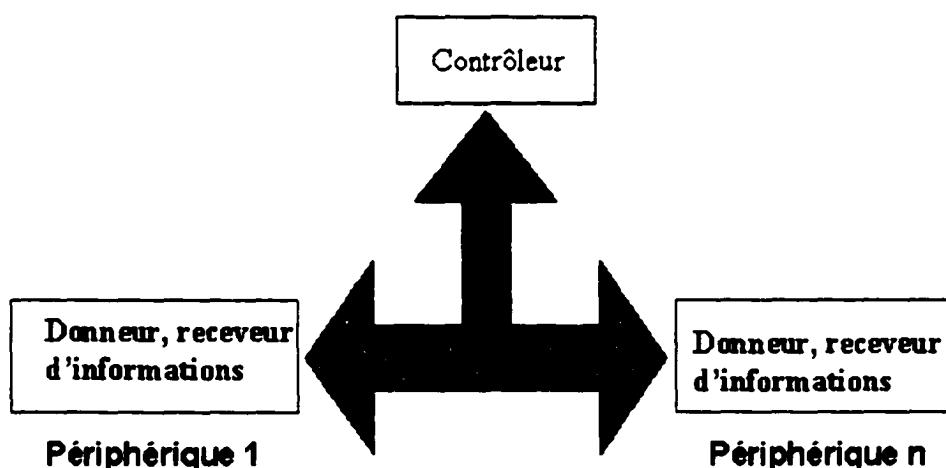


Figure 2. 6 : Le concept IEEE 488, contrôleur et périphériques.

Il existe plusieurs façons de contrôler le GPIB par logiciel :

- Utiliser un langage informatique comme C ou BASIC.
- Utiliser des langages de programmation Labwindow/CVI, qui sont des produits désignés pour communiquer directement avec le driver.

Le GPIB est donc un bus ainsi qu'un système d'interface très courant. Il est idéal pour le contrôle à distance et pour l'acquisition de mesures automatiques sur réseaux. De plus, les extenseurs du bus GPIB lui permettent d'utiliser un grand nombre d'instruments en même temps, sur des distances de plus en plus longues, en gardant toujours un taux de transfert de données important : environ 1 M.octets par seconde.

2.12 Logiciels

Pour la construction de laboratoire virtuel, il faut utiliser un logiciel qui permet de produire des applications ouvertes, faciles à personnaliser et dont l'exécution est relativement rapide. Il doit être utilisé pour la télémessure, l'acquisition de données, l'analyse, le stockage des informations et la télécommande par les réseaux.

Les langages de développement des applications de télémessure et d'apprentissage doivent être modulaire et orienté objets pour être évolutifs. Il devrait permettre le traitement multiprocesseur et l'interopérabilité entre plusieurs plate-formes.

Le rôle essentiel du logiciel est la réalisation intégrée de toutes les composantes nécessaire à un laboratoire virtuel, il utilise la diffusion en temps réel de séquences vidéo filmées ou simulées, la simulation interactive d'expériences à distance, des outils d'analyse de données, des outils de communication entre apprenant et formateur, la télémanipulation et la télémessure.

Parmi les logiciels, notons : LabVIEW (sur lequel on trouvera plus de détails en annexe A), Java et Toolbook.

Java est un environnement de programmation simple, robuste, dynamique, multi-chaîné, universel, orienté objet, indépendant de la plate-forme, créé et développé par Sun Microsystems. Les développeurs peuvent écrire des petits programmes personnalisés appelés **des applets**, qui permettent de mettre en œuvre sur plusieurs Sites Internet de nouvelle fonctionnalité : Animation, mise à jour, interaction bilatérale et plus. Quand ils sont intégrés dans les pages webs, ils permettent l'octroi de graphisme expert, l'interaction en temps réel avec les utilisateurs, la mise à jour d'information vivante et l'interaction instantanée avec des serveurs sur le réseau. Les applets Java sont

téléchargeables de n'importe quel serveur et dirigées sans risque sur n'importe quelle plate-forme et ils peuvent être incorporés dans des pages de HTML.

Toolbook est un environnement orienté objet : de nombreux outils graphiques permettent de créer des objets que l'on peut commander grâce à un langage de programmation orienté objet de haut niveau : l'Open Script.

Avec Toolbook, il est possible de créer des applications qui fonctionnent sous Windows, le programmeur peut donc accéder aux fonctionnalités communes de l'environnement Windows. Toolbook permet surtout de créer facilement un objet multimédia puisqu'on peut introduire des graphiques, des menus, des applications sonores et des clips vidéo.

Le logiciel Toolbook est un outil très intéressant puisqu'il permet de créer un livre multimédia avec une interface graphique évoluée et une interactivité réelle (grâce à la possibilité de faire des liens DDE) sans demander un lourd investissement en programmation pour le concepteur.

Comme Toolbook est un environnement orienté objet, tout ce qui apparaît à l'écran peut être considéré comme un objet. Chaque objet (fond, rectangle, cercle, page) a un ensemble de propriétés qui définissent son apparence et son comportement que l'on peut visualiser en cliquant sur l'objet en question (dans le mode concepteur).

Dans l'environnement Toolbook, le fonctionnement des applications se fait par des événements. Un événement se réalise lors de toute interaction de l'utilisateur : cela peut être par exemple par un click de souris ou un déplacement du curseur sur l'interface.

Cette architecture d'environnement, où tout est objet, où l'on procède par événement et où l'on maintient une hiérarchie, est idéale pour la programmation simple et efficace d'un logiciel multimédia interactif.

2.13 Problèmes

Bien sûr, beaucoup de problèmes sont survenus lors de l'application du projet au niveau matériel et au niveau humain.

Dès que les moyens de transmission ne sont pas dédiés à la technique de télé laboratoire, alors toutes leurs caractéristiques relatives ne sont plus contrôlables. Le débit, la vitesse et surtout le délai sont hors contrôles. Le débit et la vitesse doivent être assez haut, et le délai très court pour assurer la transmission d'un maximum de données avec le minimum de temps, ceci est, malgré les technologies de l'information et de communication disponible, assuré dans une limite très étroite. Mais à des grandes distances plusieurs questions se posent : Jusqu'à quelle mesure ces paramètres sont contrôlables ? Comment exploiter les infrastructures des réseaux à haut débit pour faciliter l'accès, la présentation et l'échange d'informations ?

En plus de ces difficultés, les buts liés au développement d'habiletés psychomotrices et ceux propres aux phases de réalisation et d'expérimentation sont les plus difficiles à atteindre dans un environnement virtuel, c'est une limitation liée à l'exercice des divers sens et à la manipulation directe. Citons à titre d'exemple ces programmes de simulation de vol sur ordinateur permettent certainement à l'utilisateur de s'initier à quelques concepts liés au vol et à des séquences d'opération mais ils ne permettent pas de développer de véritables habiletés psychomotrices utilisables aux commandes d'un appareil en vol.

Le domaine psychomoteur sera donc certainement le facteur limitatif dominant des laboratoires virtuels.

2.14 Avantages et inconvénients

Récemment, les télécommunications et les technologies de l'information sont devenues les infrastructures de l'économie des pays développés et jour après jour elles deviennent les secteurs essentiels de la formation.

Avec le concept de télé-apprentissage, un nouveau mode de vie approprié aux nouvelles technologies d'information est né : c'est aussi un nouveau monde d'apprentissage et de formation à distance qui se propage peu à peu et promet de changer toute conception reliée au temps et au lieu et mêmes aux relations humaines.

Son principal avantage, en plus de beaucoup d'autres, est d'éliminer les contraintes spatio-temporelles suite à un accès à distance multiple et rapide, surtout pour ceux qui éprouvent une difficulté de déplacement (un handicap ou autre) et par la suite une économie assez importante de temps, de coûts et d'espace d'une part et de personnel d'autre part. C'est aussi un moyen de contrôler tout milieu hostile : explosif ou électrisé, qui nécessite une haute sécurité (réactions chimiques, nucléaires et haute tension).

Ces nouvelles technologies de l'information et de la communication orientent tout travail de formation vers un travail d'échanges de connaissance et de production. Cette validation des informations sur les champs éducatifs et cette facilité de moyens de collaboration qu'offre ce concept, donnent la possibilité à une meilleure formation à des régions isolées ayant un accès normalement difficile.

Au niveau personnel, c'est un moyen qui élargit les horizons de l'imagination et de l'expression chez l'étudiant. Bien sûr, les horizons d'un tel projet ne se limitent pas au domaine d'apprentissage et de contrôle mais ça va les dépasser pour s'intégrer dans le monde industriel : les investissements pour ce type de formation auprès des entreprises sont très importants. Un nouveau concept de travail va débiter avec le début du 21^{ème} siècle.

Comme tout système, celui-ci présente aussi bien des avantages que des inconvénients.

Ces inconvénients s'étendent aussi à tous les niveaux, surtout humain : notre dépendance de la technologie et de la machine devient de plus en plus grande. Toute activité psychomotrice est déjà atténuée : on se colle au siège de plus en plus et le mouvement devient hors sujet du travail quotidien. Ceci, un peu plus développé, va bien sûr conduire à des résultats nuisibles à la santé et à la mentalité puisque les contacts directs Humain- Humain diminuent de plus en plus et une génération se forme en ayant contact avec la machine. En conclusion, de nos jours la technologie se développe au détriment de la pédagogie. Et quelle formation attendions-nous ??

2.15 Statistiques

Diverses études s'accordent pour dire que la formation par les nouveaux médias d'apprentissage (NMA) permet de réduire 25% à 50% le temps consacré à l'apprentissage et de 75% les coûts liés au transport et à l'hébergement des apprenants. Pour les années prochaines, on estime à 1 milliard de dollars les ventes des programmes de télé apprentissage multimédias. Selon "Industrie Canada ", des économies annuelles de l'ordre de 223\$ par employé sont prévues d'ici l'an 2005 dans les secteurs qui font appel aux NMA, soit une augmentation de la productivité de la main d'œuvre de 0.5%

attribuable à cette technique, représentant des économies annuelles de 2 milliards de dollars pour l'économie canadienne. Cette réduction des coûts pourrait être à l'origine de 9500 employés d'ici l'an 2005 [10].

2.16 Conclusion

Dans ce chapitre, on a fait un survol sur l'aspect du laboratoire virtuel, ses éléments et ses outils. Sauf qu'on doit indiquer que le matériel et les appareils de notre laboratoire sont réels malgré que son nom indique le contraire, pour le faire distinguer à d'autres laboratoires qu'on peut trouver sur Internet et qui ont des éléments virtuels. Dans notre cas, le laboratoire existe réellement et physiquement même les matériels, on va ajouter seulement l'accès à distance pour pouvoir faire des tests et des mesures. Il s'agit bien de la télé-contrôle. Il faut donc dire qu'on est en train de rendre notre système éducatif plus sensible et plus délicat et même plus fragile. Il faut assurer alors la sécurité et la fiabilité de ce système ainsi que sa gestion.

CHAPITRE 3

COMMANDE À DISTANCE D'UN MOTEUR À COURANT CONTINU EN TEMPS RÉEL

3.1 Introduction

Dans ce chapitre, nous présenterons une étude théorique sur les principaux outils mathématiques utilisés dans la représentation temporelle et fréquentielle des signaux et des systèmes physiques rencontrés dans l'industrie, apprendrons les techniques de modélisation mathématiques, d'analyse et de simulation des systèmes physiques.

Nous trouverons aussi une présentation sur l'identification et la modélisation des systèmes dynamiques, notion de base, notion de système, modélisation et construction de système et leur classification mathématique. On explique l'asservissement et la régulation, les qualités d'un asservissement et les rôles des régulateurs, puis la conception d'encodeur, son type et le principe de fonctionnement.

Nous expliquerons la conception du régulateur en position PD, la simulation et le comportement du moteur, le principe de fonctionnement et la modélisation de moteur à courant continu.

À la fin nous présenterons la méthode suivie pour asservir le moteur avec les outils nécessaires pour le contrôler à distance à partir d'un micro ordinateur, ainsi que les résultats obtenus. Les programmes avec les interfaces utilisés seront aussi présentés.

3.2 Généralités

La commande et l'interprétation du comportement de procédés industriels ou de phénomènes physiques naturels font partie des tâches qui incombent à l'ingénieur. Ce dernier est confronté à une réalité qu'il lui faudra domestiquer et/ou comprendre pour en tirer la meilleure partie.

Au centre de cette connaissance se trouve le concept de système, concept que l'on retrouve dans un grand nombre de disciplines et techniques : Contrôle de procédés, techniques d'optimisation, traitement du signal, filtrage, mathématique des équations différentielles, algèbre linéaire, etc....

Dans le cadre de ce projet, nous allons nous intéresser principalement au domaine de la commande simple, terme que nous pouvons définir de la façon suivante : L'action de commander consiste à appliquer à l'entrée d'un système un signal de consigne, une sollicitation, en vue d'obtenir de sa part une réponse déterminée.

Nous nous intéresserons essentiellement à la technique dite des *systèmes asservis* qui consiste à comparer les signaux des consignes et de sortie. Cette dernière étant obtenue à l'aide d'un appareil de mesure (capteur). Voir figure 3.1.

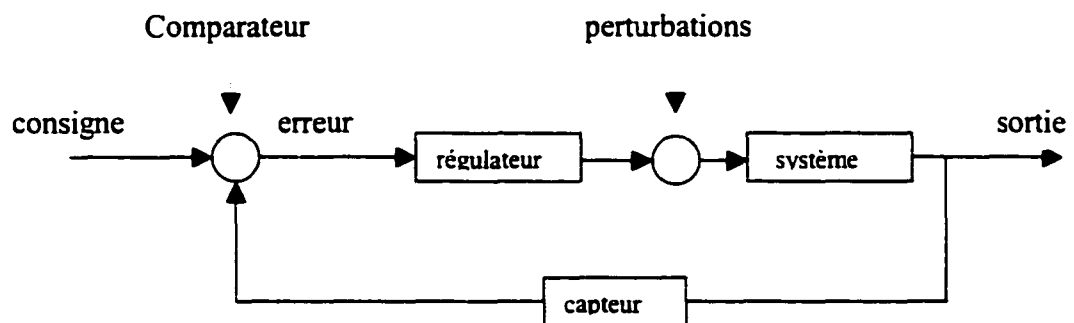


Figure 3. 1 : Représentation générale.

Les domaines d'application de la commande des systèmes sont nombreux. Certains peuvent même être qualifiés de spectaculaires, en particulier l'aéronautique (commandes électriques d'avions instables, optimisation de trajectoires), la conquête spatiale (guidage de lanceurs, stabilisation de satellites) ou la robotique (commande de bras manipulateur, guidage de robots...). Bien d'autres applications peuvent être répertoriées (contrôle de vibrations de rotors d'alternateur ou de rotors d'hélicoptères, contrôle de vitesse et de position de bras dans un disque magnétique, positionnement de caméra, etc....).

Les moteurs à courant continu asservis en position (ou en vitesse) sont utilisés dans de très nombreuses applications : Pilote automatique de bateau, volets de freinage sur les ailes des avions, systèmes de pointage des télescopes, outils tels que perceuse, ponceuse....

3.3 Identification et Modélisation des systèmes dynamiques

Les signaux et les systèmes, figure 3.2, sont des notions de base à connaître pour aborder le domaine des asservissements. Les signaux que nous envisageons sont présents en entrée (consignes, bruits ...). et en sortie d'un procédé physique opérant une transformation sur les signaux d'entrée :

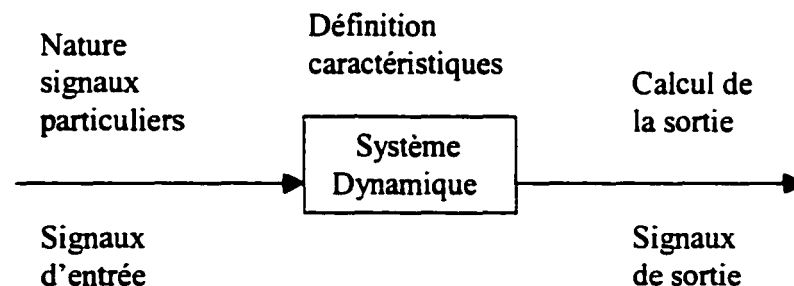


Figure 3. 2 : Notions d'un système dynamique.

Pour étudier un système dynamique il faut d'abord commencer par :

- Préciser la nature des signaux aléatoires ou déterministes, à temps discret ou à temps continu (transformée de Laplace, transformée de Fourier et transformée en z), choisir un modèle et définir des cas particuliers.
- Noter les caractéristiques du système telles que nature (numériques, à temps continu, etc...), linéarité, invariance, et à choisir un modèle fonctionnel, une fonction de transfert, explicitant la transformation entrée-sortie mise en œuvre.
- Choisir l'outil mathématique le mieux adapté pour calculer la sortie en fonction de l'entrée ou mettre en évidence des propriétés caractérisant le comportement du système.

3.3.1 Notion de Système

Un système est un ensemble d'éléments associés pour atteindre un but déterminé au moyen d'un fonctionnement spécifié. Un système est considéré comme séparé du milieu ambiant et d'autres systèmes extérieurs par une surface imaginaire qui coupe la liaison entre eux et le système considéré. Par cette liaison, le système subit les actions d'ambiance ou celles des systèmes extérieurs ou bien agit lui-même sur le milieu ambiant ou les systèmes extérieurs.

Nous pouvons caractériser un système par des relations entrées et sorties. Le système peut être un amplificateur contenant des composants électroniques, un modulateur de largeur d'impulsion, un procédé chimique contenant plusieurs processus chimiques.

Exemples : système solaire, machine à pâte et papier, etc....

3.3.2 Processus (procédé)

Dans un processus, plusieurs variables interagissent et produisent des variables observables. Les variables observables d'intérêt sont appelées sorties. Le processus est aussi affecté par des variables externes, comme nous montre la figure 3.3, donc celles qu'on peut manipuler, les entrées du processus, les perturbations. Nous pouvons distinguer les perturbations mesurables que nous pouvons directement mesurer et les perturbations non mesurables dont nous pouvons seulement observer l'influence sur les sorties.

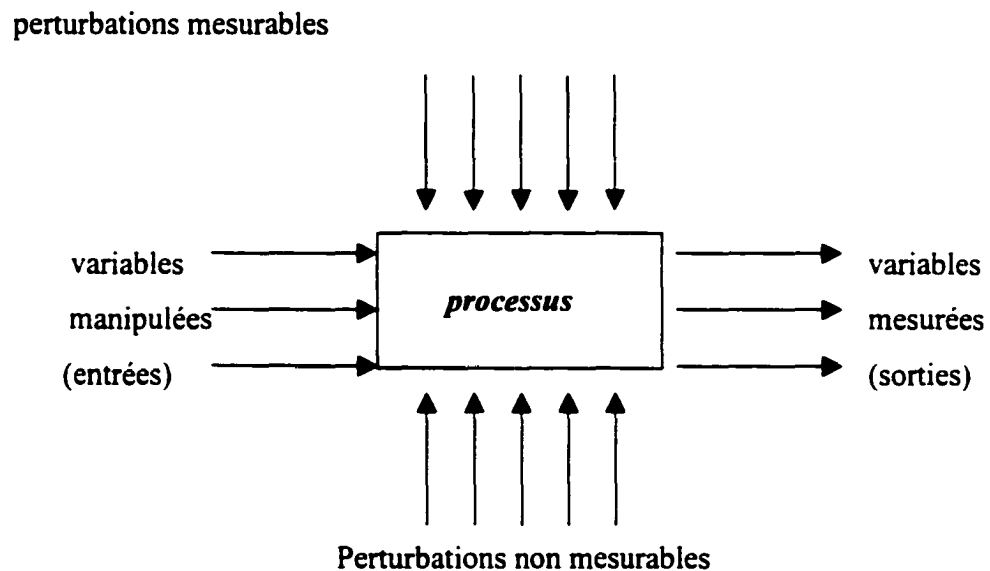


Figure 3. 3 : Effets de perturbations.

3.3.3 Modélisation d'un système

La modélisation est l'art de relier des observations sur un système à un moule donné. C'est aussi l'art d'extraire les aspects essentiels d'un système à partir d'un environnement turbulent et parfois même chaotique et de les traduire dans un langage abstrait et précis en une forme utilisable que nous pouvons appeler **modèle**. Un modèle

ne devrait pas être compliqué puisqu'il serait difficile de l'utiliser pour prédire le comportement du système ni trop simplifié afin que les prédictions ne soient pas imprécises.

Le développement d'un modèle pour un système physique peut être réalisé pour différentes raisons; le dimensionnement d'une installation, la formation des opérateurs ou la compréhension des phénomènes. Principalement, nous utilisons un modèle pour répondre à nos questions sur le système sans avoir à expérimenter physiquement sur le système.

La notion de modèle est fondamentale. En effet, il existe une grande variété de modèles pouvant décrire le même procédé, chacun d'eux ayant évidemment une finalité différente. Ici on cherche à avoir un modèle complet qui décrit bien le système afin de faciliter l'étape de synthèse et valider le régulateur conçu et permettre de vérifier si ce régulateur répond aux spécifications.

L'identification comporte nécessairement une étape expérimentale et on peut décomposer un travail d'identification en quatre étapes distinctes :

- L'envoi des signaux tests vers le processus et l'acquisition de la réponse.
- Le calcul des paramètres du système.
- La construction d'un modèle qui doit décrire le système.
- La comparaison des sorties du processus et du modèle pour validation.

3.4 Modèles mathématiques

Les modèles mathématiques sont des relations pour effectuer l'analyse et la synthèse d'un système dynamique et connaître les relations entre ses grandeurs d'entrée et ses grandeurs de sortie. Ces modèles sont particulièrement nécessaires pour

l'application de la théorie moderne de systèmes et de commande, mais sont utilisés aussi dans les domaines du génie pour fin de prédiction et de simulation.

3.4.1 Classification des modèles mathématiques

Les modèles mathématiques qui sont développés pour différents systèmes peuvent avoir des caractéristiques et des propriétés différentes.

Nous pouvons distinguer deux types de modèle :

1- Modèle statique :

C'est un système qui décrit la relation entre l'entrée et la sortie comme une fonction c'est à dire que la sortie à l'instant « t » dépend seulement de l'entrée. Ces modèles n'ont pas de mémoire et décrivent un état permanent du système. Ils pourraient être utilisés pour :

- La recherche du point de fonctionnement.
- Le dimensionnement d'une installation.

2- Modèle dynamique :

Ces modèles permettent de représenter l'évolution d'un système dans le temps, ce sont des modèles avec mémoire c'est à dire que la sortie à l'instant « t » dépend de l'entrée à l'instant « t » et des temps précédents.

3.5 Construire un modèle

Pour construire un modèle, il faut utiliser deux sources d'information : l'information à priori sur le système issue de l'expérience acquise à partir de la modélisation d'autres processus et de la littérature accumulée au cours des années et permettant d'élaborer un modèle dit **modèle de connaissance**.

L'autre source d'information est à posteriori et se base sur les observations sur le système lui-même. Elle permet d'élaborer un modèle dit de comportement ou **boîte noire**.

3.5.1 Modélisation physique (modèle de connaissance)

Le principe consiste à découper les propriétés du système en sous-systèmes dont on connaît le comportement. Pour les systèmes techniques, il s'agit d'écrire les lois de la physique et de la chimie (équilibre de masse et d'énergie) régissant ces sous-systèmes pour obtenir les équations mathématiques reliant les différentes variables. Pour les systèmes non techniques les lois de la physique ne sont pas disponibles, il faut utiliser des hypothèses ou des faits bien connus.

3.5.2 Identification (modèle de comportement ou boîte noire)

L'identification des systèmes consiste à construire un modèle mathématique en utilisant les observations et les données expérimentales sur le système et d'ajuster les propriétés de ce modèle. Ce modèle est généralement linéaire et sa validité reste limitée à de petites variations autour du point de fonctionnement. Trois composants constituent l'identification : Les données, la structure du modèle et le critère d'estimation du modèle.

3.6 Asservissement et régulation

3.6.1 Asservissement

Un système asservi est un système dit suiveur, c'est la consigne qui varie : exemple ; une machine outil qui doit usiner une pièce selon un profil donné, un missile qui poursuit une cible.

Le but du système asservi à feedback est d'obtenir un signal de sortie relié au signal d'entrée par une relation quelconque avec la précision requise. Cette tâche ne constitue qu'une partie de l'objectif du régulateur.

Le régulateur devra être calculé de façon à respecter un certain nombre de contraintes et le système doit compenser l'effet des perturbations. Nous pouvons schématiser celles-ci de la façon suivante :

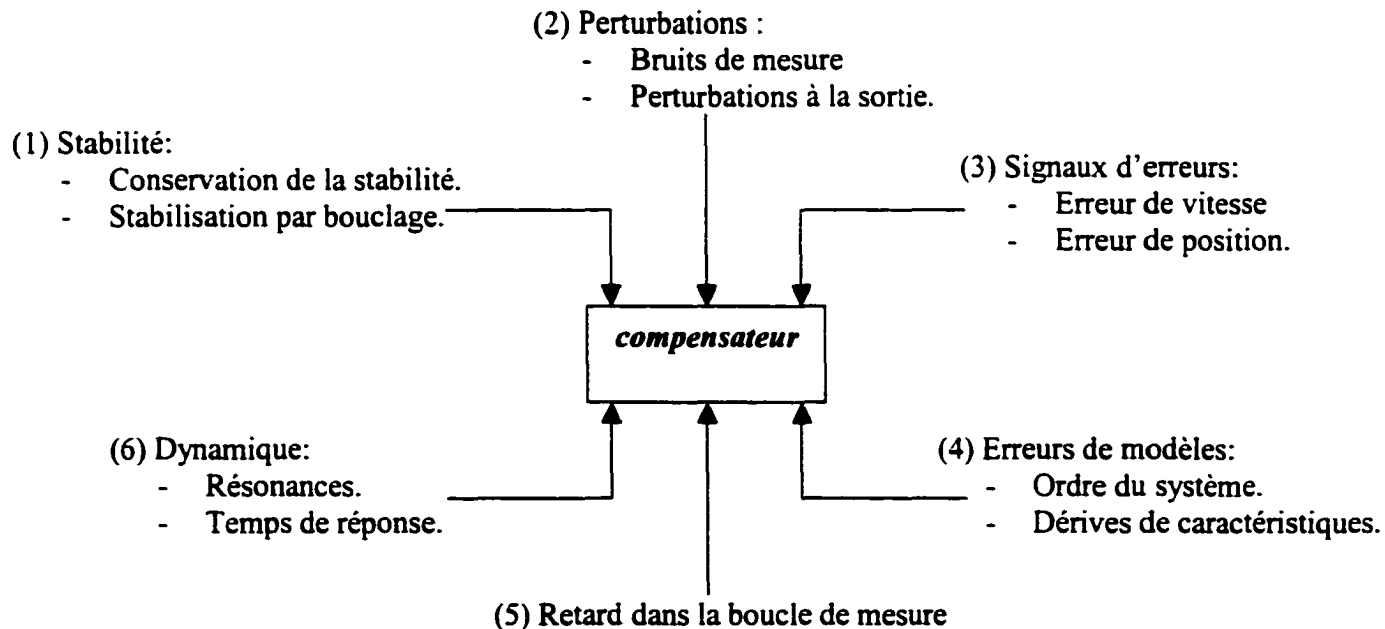


Figure 3. 4 : Contraintes prises en compte dans la synthèse.

3.6.2 Régulation

La plupart des régulations industrielles sont effectuées par les modes d'action continue où, à chaque valeur de l'écart à l'intérieur de la limite de régulation, correspond une valeur déterminée et unique, du signal de sortie du régulateur.

Le signal de sortie peut prendre toutes les valeurs comprises entre la portée maximale et la portée minimale. Le mode d'action continue, qui peut imposer au signal de sortie une variation continue entre deux limites déterminées est appelé progressif.

L'intelligence ou la perception des régulateurs à action continue va au-delà du seul signe de l'écart ($x - x_0$) et selon cette perception les modes en continu sont classés en trois catégories principales :

- 1- L'action proportionnelle P.
- 2- L'action intégrale I.
- 3- L'action dérivée D.

Chacun de ces modes se distingue par la manière dont le signal affecte le signal de sortie, mais d'une façon générale les régulateurs sont à action combinée de deux ou trois des actions P, I et D.

L'objectif du système de régulation est l'élimination des influences des grandeurs perturbations sur la variable régulée du procédé.

À titre d'exemple : Le réglage de la température dans un four, de la pression dans un réacteur et le niveau d'eau dans un réservoir.

3.7 Qualités d'un asservissement

L'organe de commande doit à la fois préserver les qualités du système et permet d'assurer les performances satisfaisantes. Comme l'illustre la figure 3.5, nous allons envisager ainsi tour à tour les points suivants :

- La stabilité du système compensé.
- La précision ou la capacité du système à suivre une consigne particulière sans erreur.
- L'aptitude du système à limiter les effets des perturbations sur la sortie ou des bruits de mesure.
- Les caractéristiques dynamiques lorsqu'on soumet le système à des entrées particulières le plus souvent l'échelon unité avec temps de réponse et le dépassement autorisé.

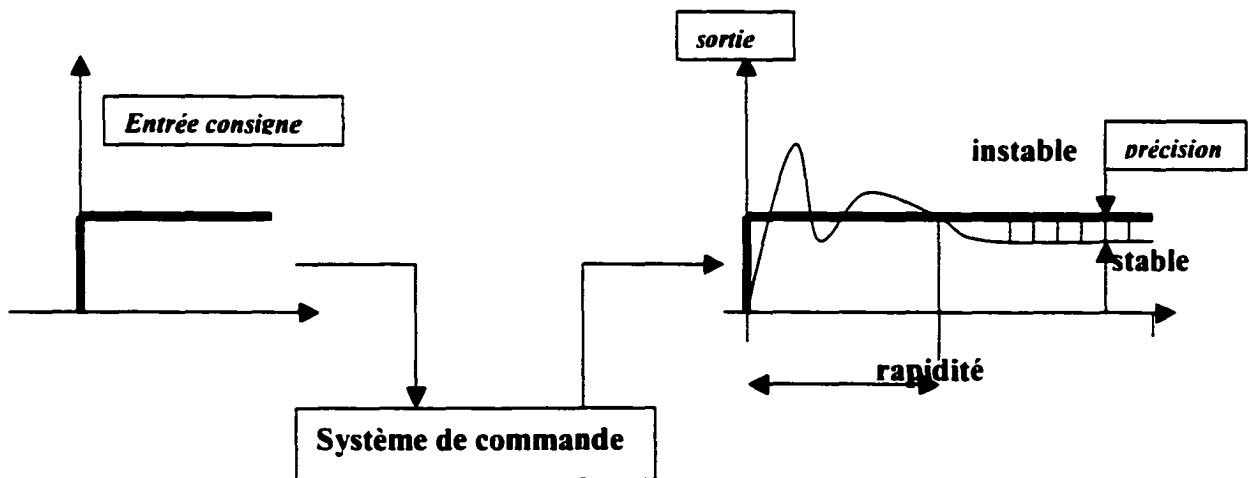


Figure 3. 5 : Système de commande.

3.8 Contrôleur (Régulateur)

Un contrôleur ou régulateur, figure 3.6, est un instrument qui est utilisé en conjonction avec d'autres dispositifs pour contrôler automatiquement une variable de procédé à un point de consigne prédéterminé.

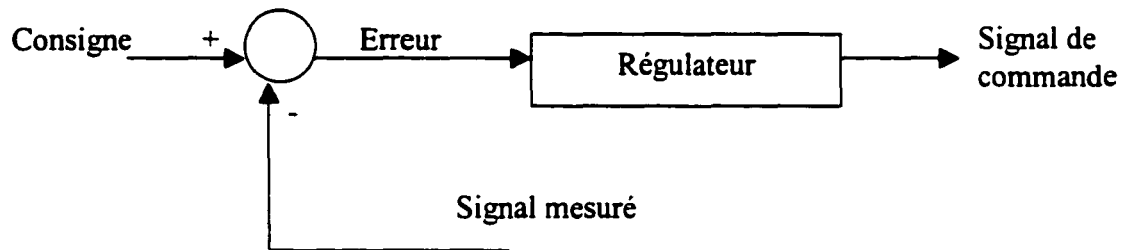


Figure 3. 6 : Représentation générale d'un régulateur.

Le contrôleur compare la valeur de la variable du procédé avec la valeur de la consigne et dérive un signal d'erreur. Puis il produit un signal de sortie qui est une fonction du signal d'erreur d'entrée. La relation entre le signal de sortie du contrôleur et le signal d'entrée du contrôleur définit le mode de contrôle.

Les modes de contrôle les plus utilisés ou communs sont les suivants :

- 1- Contrôleur de type proportionnel.
- 2- Contrôleur de type intégral.
- 3- Contrôleur de type dérivé.

Le régulateur le plus complet à ce titre est donc le régulateur PID (ses initiales signifient Proportionnel Intégral Dérivé et indiquent que toutes ces actions sont présentes dans ce régulateur). A partir de ce régulateur, on a des régulateurs plus

spécifiques, le régulateur PI (Proportionnel Intégral), le régulateur PD (Proportionnel Dérivé) ou simplement le régulateur P (Proportionnel).

3.8.1 Régulateur PID

Un régulateur PID, montré dans la figure 3.7, peut être utilisé pour pouvoir contrôler un système avec précision. En combinant les trois actions de base, il permet de bénéficier des avantages du régulateur PD comme de ceux du PI. Ainsi, on pourra améliorer la précision, une propriété que l'on ne peut pas avoir avec le régulateur PD, tout en ayant de la latitude pour régler la marge de phase, une latitude que le régulateur PI ne permet pas.

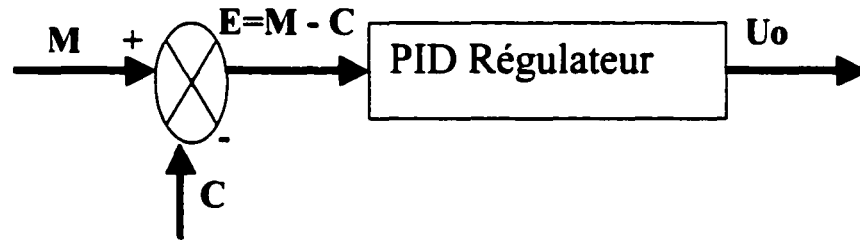


Figure 3. 7 : Régulateur PID.

La fonction de transfert d'un correcteur PID est donnée par :

$$U = K_p * e + \frac{K_p}{T_i} \int_0^t e dt + K_p T_d \frac{de}{dt} + U_0 \quad (3.1)$$

$$U = K_p E(s) + \frac{K_p}{T_i s} E(s) + K_p T_d s E(s) = K_p \left(\frac{1 + T_i s + T_i T_d s^2}{T_i s} \right) E(s) \quad (3.2)$$

Où k_p est le gain de l'action P, T_i la constante de temps de l'intégrateur, T_d celle de l'action D.

L'action proportionnelle permet d'améliorer la rapidité de la réponse mais introduit de l'instabilité. Elle corrige de manière instantanée, donc rapide, tout écart de la grandeur à régler, elle permet de vaincre les grandes inerties du système. Afin de diminuer l'écart de réglage et rendre le système plus rapide, on augmente le gain (on diminue la bande proportionnelle) mais, on est limité par la stabilité du système. L'action P est utilisée lorsqu'on désire régler un paramètre dont la précision n'est pas importante, exemple : régler le niveau dans un bac de stockage. La figure 3.8 montre les rôles de chaque action.

L'action intégrale permet d'augmenter la précision aux basses fréquences et corrige l'erreur statique, elle complète l'action proportionnelle et permet d'éliminer l'erreur résiduelle en régime permanent. Afin de rendre le système plus dynamique (diminuer le temps de réponse), on diminue l'action intégrale mais ceci provoque l'augmentation du déphasage ce qui provoque l'instabilité en boucle fermée.

Réponse indicielle

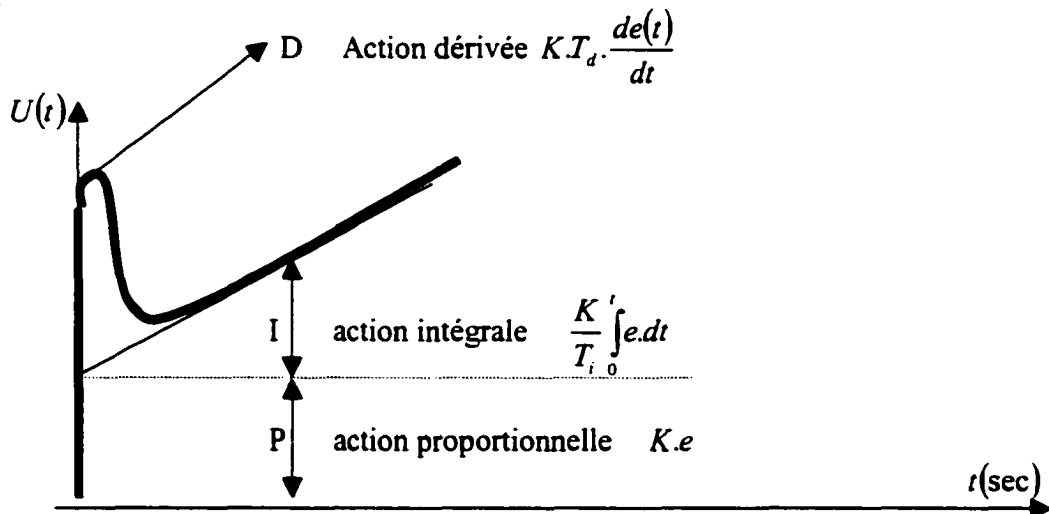


Figure 3. 8 : Rôle des actions P, I et D.

L'action intégrale est utilisée lorsqu'on désire avoir en régime permanent, une précision parfaite, en outre, elle permet de filtrer la variable à régler d'où l'utilité pour le réglage des variables bruitées telles que la pression

L'action dérivée permet d'augmenter la marge de stabilité aux hautes fréquences, elle permet aussi, en compensant les inerties dues au temps mort, d'accélérer la réponse du système et d'améliorer la stabilité de la boucle, en permettant notamment un amortissement rapide des oscillations dues à l'apparition d'une perturbation ou à une variation subite de la consigne.

Dans la pratique, l'action dérivée est appliquée aux variations de la grandeur à régler seule et non de l'écart mesure-consigne afin d'éviter les à-coups dus à une variation subite de la consigne. Elle est utilisée dans l'industrie pour le réglage des variables lentes telles que la température, elle n'est pas recommandée pour le réglage d'une variable bruitée ou trop dynamique (la pression). En dérivant un bruit, son amplitude risque de devenir plus importante que celle du signal utile.

Afin d'augmenter les performances d'un système, on peut agir sur les paramètres du régulateur pour assurer les valeurs optimums selon un critère choisi. Mais, quelquefois l'algorithme de régulation utilisé ne peut plus assurer la stabilité ou les bonnes performances car le système à commander présente par exemple un temps de retard important, alors on est contraint d'agir sur la structure de la boucle de régulation (cascade, compensation etc.) ou d'utiliser des algorithmes plus performants (retour d'état, optimal, découplage etc.)

Les formes les plus utilisées de ces régulateurs sont :

- 1- Régulateur parallèle
- Avec une référence variable

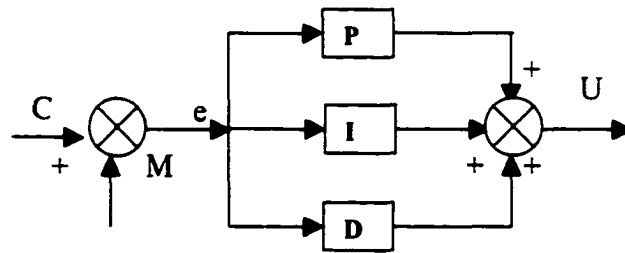


Figure 3. 9 : Régulateur parallèle avec référence variable.

$$U = Ke + \frac{K'}{T_i} \int_0^t e dt + KT_d \frac{de}{dt} + U_0 \quad (3.3)$$

- Avec une référence constante

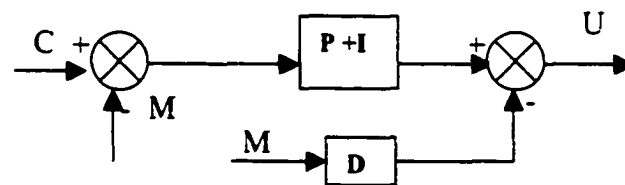


Figure 3. 10 : Régulateur parallèle avec référence constante.

$$U = Ke + \frac{K'}{T_i} \int_0^t e dt - KT_d \frac{dM}{dt} + U_0 \quad (3.4)$$

2- Régulateur mixte

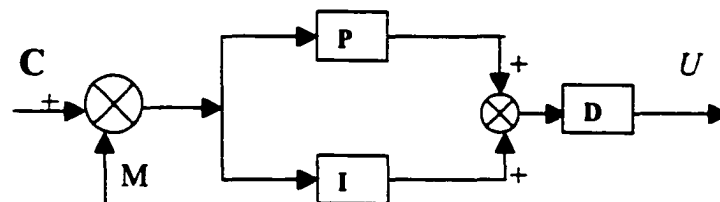


Figure 3. 11 : Régulateur mixte.

$$U = \frac{K.K_0(1 + T_i s + T_i T_d s^2)}{(1 + (\tau_1 + \tau_2)p + \tau_1 \tau_2 s^2)T_i s} \quad (3.5)$$

Ces régulateurs sont à usage très général (on peut régler un très grand nombre de processus à partir d'un PID ou d'un PI) mais il existe aussi des régulateurs qui ont un rôle bien particulier : Les régulateurs à Avance de phase ou les régulateurs à Retard de phase.

3.9 Boucle de contrôle

Il y a deux types de base de boucle de contrôle :

3.9.1 Boucle de contrôle ouverte

Aucune information venant du procédé n'est fournie au contrôleur. Ainsi la sortie de procédé n'affecte pas la sortie du contrôleur. Figure 3.12.



Figure 3. 12 : Système boucle ouverte.

3.9.2 Boucle de contrôle fermée

Dans le contrôle de boucle fermée, la sortie du procédé est retournée au contrôleur et est utilisée pour ajuster automatiquement la sortie de contrôleur. Figure 3.13.

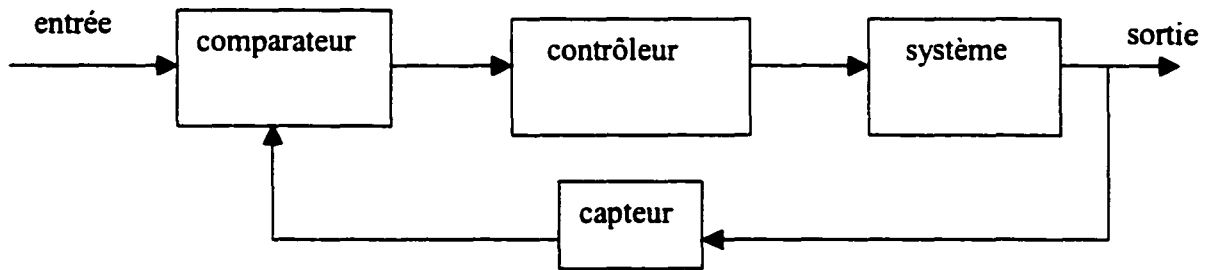


Figure 3. 13 : Système boucle fermée.

3.10 Moteur à courant continu

Les machines à courant continue peuvent fonctionner indifféremment soit comme réceptrices (moteurs), soit comme génératrices (dynamos). Figure 3.14 et 3.15.

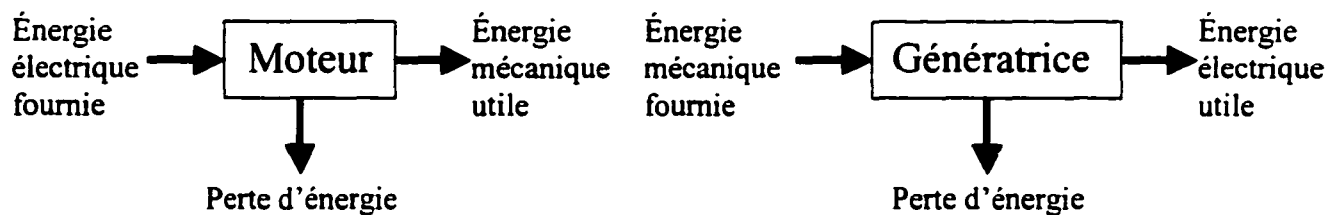


Figure 3. 14 : Conversion d'énergie.



Figure 3. 15 : Symbole.

En effet, bien que la consommation de l'énergie en courant continu soit de plus en plus élevée (commande des moteurs à vitesse variables...) cette énergie n'est pas

produite par des machines tournantes car on préfère utiliser des redresseurs statiques à diodes ou à thyristors.

Les réceptrices en revanche, sont très utilisées. Le moteur à courant continu étant le moteur idéal pour les entraînements à vitesse variable.

Une machine courant continu est constituée d'un circuit inducteur au stator et d'un circuit induit au rotor.

3.10.1 Les éléments constitutifs du moteur

Un moteur, comme on peut voir dans la figure 3.16, comprend les éléments suivants :

- Un **circuit magnétique** comportant une partie fixe, le stator, une partie tournante, le rotor et l'entrefer l'espace entre les deux parties.

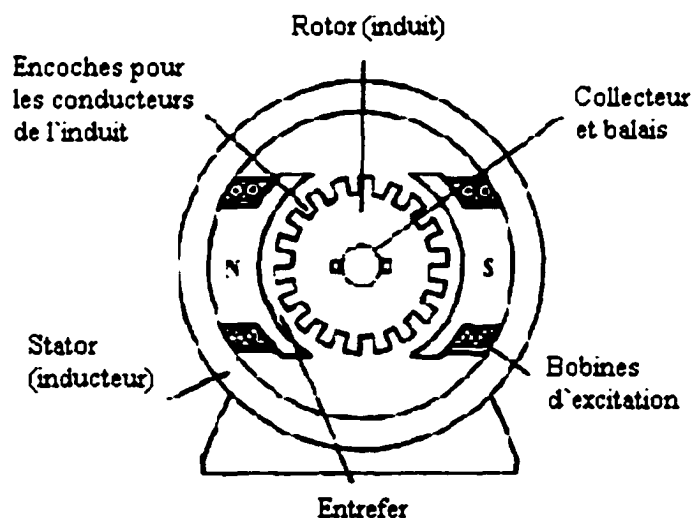


Figure 3. 16 : Un moteur.

- Une source de champ magnétique nommée ***l'inducteur*** (le stator) créée par un bobinage ou des aimants permanents.
- Un circuit électrique ***induit*** (le rotor) subit les effets de ce champ magnétiques.
- Le ***collecteur*** et les ***balais*** permettent d'accéder au circuit électrique rotorique.

Les pôles inducteurs génèrent dans l'entrefer une distribution d'induction fixe dans l'espace correspondant à une onde de potentiel magnétique et présentant des pôles alternativement Nord-Sud. L'interaction entre les champs magnétiques de l'inducteur et de l'induit se traduit par un couple électromagnétique. Normalement les balais sont calés de telle sorte que l'axe magnétique de l'enroulement pseudo-stationnaire constitué par l'enroulement d'induit coïncide avec l'axe neutre.

3.10.2 Le fonctionnement du moteur

Les moteurs à courant continu à excitation séparée présentent des caractéristiques, en particulier de couple moteur, qui les rendent attrayants pour la réalisation de servomécanismes. Nous avons deux façons de commander un moteur à C.C:

- 1- Le moteur à commande d'induit (armature).
- 2- Le moteur commandé par le champ.

La machine fonctionne en moteur si l'induit est alimenté ou en génératrice si le rotor est entraîné à une certaine vitesse angulaire. Figure 3.17.

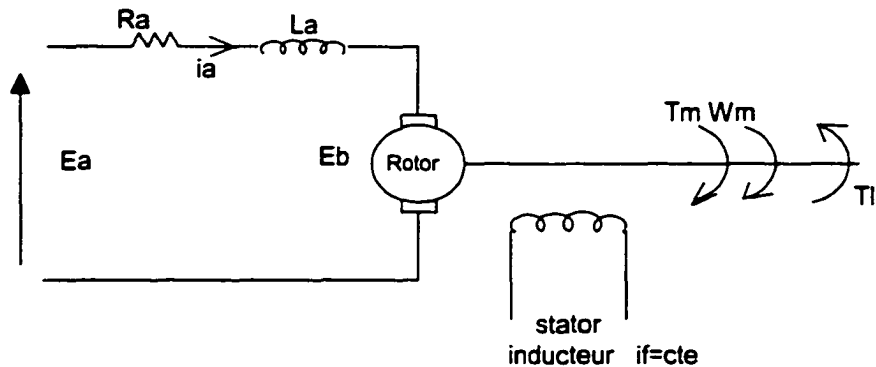


Figure 3. 17 : Schéma d'un moteur à CC.

Les variables sont définies comme suit :

- E_a = Tension appliquée (volts)
- i_a = Courant dans le rotor (A)
- R_a = Résistance d'induit (moteur) (Ω)
- L_a = Inductance d'induit (H)
- E_b = Tension induite (volts)
- T_m = Couple moteur (N.m.)
- T_l = Couple charge (N.m.)
- ω_m = Vitesse de rotation du moteur (rad/s.)

Le moteur à courant continu, figure 3.18, fait intervenir trois phénomènes :

- | | | |
|---------------------|----------------|---------------------|
| • <i>Électrique</i> | entrée : V_a | sortie : i_a |
| • <i>Magnétique</i> | entrée : i_a | sortie : T_m |
| • <i>Mécanique</i> | entrée : T_m | sortie : ω_m |



Figure 3. 18 : Schéma de trois phénomènes dans un moteur à CC

a) Partie électrique

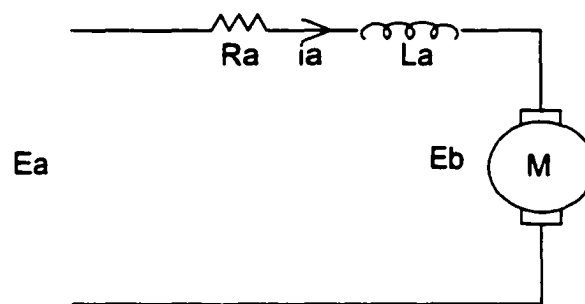


Figure 3. 19 : Phénomène électrique au moteur à CC.

Le voltage aux bornes de l'armature est donné par l'équation suivante :

$$E_a = R_a I_a + L_a \frac{dI_a}{dt} + E_b \quad (3.6)$$

$$E_b = K_m \Phi \omega_m \quad (3.7)$$

Où K_m : constante de vitesse du moteur (volt.sec/rad)

Φ : flux inducteur constant.

Dans cette équation, la tension force contre-électromotrice du moteur est :

$$E_b = K_b \omega_m \quad (3.8)$$

Où K_b : constante de la force contre-électromotrice du moteur (volt.sec/rad)

b) Partie magnétique

L'intensité du champ magnétique du moteur étant constante, le courant qui s'écoule dans l'induit produit un couple qui lui est proportionnel.

$$T_m = K_m \varphi I_a \quad (3.9)$$

$$= K_T I_a \quad (3.10)$$

où $K_T =$ constante du couple (N.m/A)

En pratique : $K_t = K_b$.

c) Partie mécanique

En appliquant la deuxième loi de Newton :

$$T_m = J_m \frac{d\omega}{dt} + B_m \omega + T_L \quad (\text{coté moteur}) \quad (3.11)$$

Où :

T_m : couple (N.m)

J_m : inertie (kg.m² ou N .m sec²/rad)

B_m : coefficient de frottement du moteur (kg.m ou N.ms/rad)

T_L : charge extérieure appliquée sur le moteur (N.m)

Si on applique la loi de Laplace sur les trois équations avec les conditions initiales nulles, on obtient les relations suivantes :

1- Équations électriques

$$E_a(s) = R_a I_a(s) + L_a(s) I_a(s) + E_b(s) \quad (3.12)$$

$$E_a(s) = I_a(s) (R_a + L_a s) + E_b(s) \quad (3.13)$$

$$I_a(s) = \frac{Ea(s) - Eb(s)}{Ra + Las(s)} \quad (3.14)$$

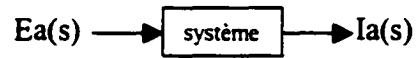


Figure 3. 20 : Schéma bloc de système électrique.

2- Équations magnétiques

$$Eb(s) = Kb \Omega m(s) \quad (3.15)$$

$$Tm(s) = Kt Ia(s) \quad (3.16)$$

$$Tm = KbI_a \quad (3.17)$$

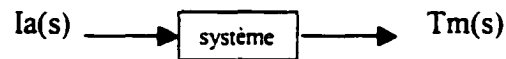


Figure 3. 21 : Schéma bloc de système magnétique.

3- Équations mécaniques

$$Tm(s) = Jms\Omega m(s) + Bm\Omega m(s) + Tl(s) \quad (3.18)$$

$$Tm(s) = (Jms + Bm)\Omega m(s) + Tl(s) \quad (3.19)$$

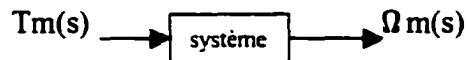


Figure 3. 22 : Schéma bloc de système mécanique.

On peut trouver la relation entre la sortie Ω_m et l'entrée T_m :

$$\Omega_m(s) = \frac{T_m(s) - T_L(s)}{J_m + B_m} \quad (3.20)$$

3.10.3 Construction du schéma-bloc

Les équations ci-dessus nous donne le schéma bloc de la figure 3.23.

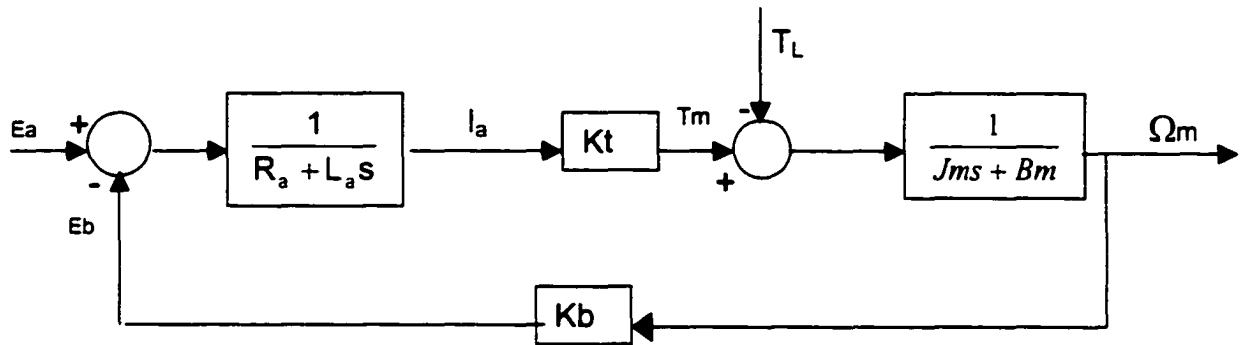


Figure 3. 23 : Schéma bloc du moteur c.c.

Cependant, la fonction de transfert du moteur, généralement considérée lorsque $T_L = 0$ est :

$$\frac{\omega_m(s)}{E_a(s)} = \frac{\frac{1}{(R_a + L_a s)} * K_t * \left(\frac{1}{J_m s + B_m} \right)}{1 + \left(\frac{1}{R_a + L_a s} \right) * \left(\frac{1}{J_m s + B_m} \right) * K_b^2} \quad (3.21)$$

Après simplification :

$$\frac{\omega_m(s)}{E_a(s)} = \frac{K_b}{J_m L_a s^2 + (J_m R_a + B_m L_a) s + K_b^2 + R_a B_m} \quad (3.22)$$

En pratique, la valeur (La) est négligée ce qui nous donne un moteur modélisé par un système du premier ordre comme suit :

$$\frac{\omega_m(s)}{E_a(s)} = \frac{K_t}{J_m R_a s + K_t K_b + R_a B_m} \quad (3.23)$$

$$= \frac{K_b}{K_b^2 + R_a B_m} * \frac{1}{1 + \left(\frac{J_m R_a}{K_b^2 + R_a B_m} \right) s} \quad (3.24)$$

Finalement, cette dernière relation nous donne :

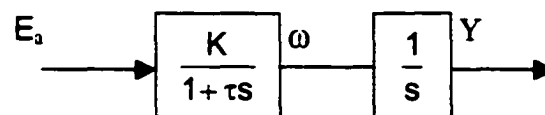
$$\frac{\omega_m(s)}{E_a(s)} = \frac{K}{1 + \tau s} \quad (3.25)$$

Avec :

K = gain statique du moteur (rad/sec.v)

τ = constante de temps du moteur

Le moteur est modélisé comme un système du premier ordre pour l'asservissement de vitesse, ce qui donne en position la boucle ouverte suivante :



Avec

K = gain statique du moteur = $1/K_m$

τ = constante du temps du moteur = $(R_a * J / K_m^2)$.

$$\frac{Y(s)}{E_a(s)} = \frac{K}{\tau s^2 + s} \quad (3.26)$$

On a un système du second ordre, qui a deux pôles réels.

3.11 L'encodeur

Le contrôle du déplacement et de la position d'un mobile est un problème couramment rencontré sur un grand nombre de systèmes automatisés. L'encodeur rotatif est un capteur de position angulaire. Lié mécaniquement à un arbre qui l'entraîne, son axe fait tourner un disque qui lui est solidaire. Ce disque comporte une succession de parties opaques et transparentes.

Une lumière émise par des diodes électrolumineuses (DEL) traverse les fentes de ce disque et crée sur les photodiodes réceptrices un signal analogique. Une interface électronique (qui est incluse dans l'encodeur) amplifie ce signal puis le convertit en signal carré qui est alors transmis à un système de traitement.

Il existe deux types d'encodeur de position rotatifs :

- L'encodeur incrémental.
- L'encodeur absolu.

L'annexe B donne une brève description de ces deux types d'encodeurs. Dans notre application l'encodeur incrémental a été utilisé.

3.12 Les éléments du système

Le système au complet, montré dans la figure 3.24 comporte plusieurs éléments tels que : Un micro-ordinateur, un moteur à courant continu (Annexe C), lié à un ascenseur qui représente la charge, une carte d'acquisition entrées-sorties (Annexe D), un encodeur optique (Annexe C). En annexe E, on trouve les E/S connexions pour la carte d'acquisition.

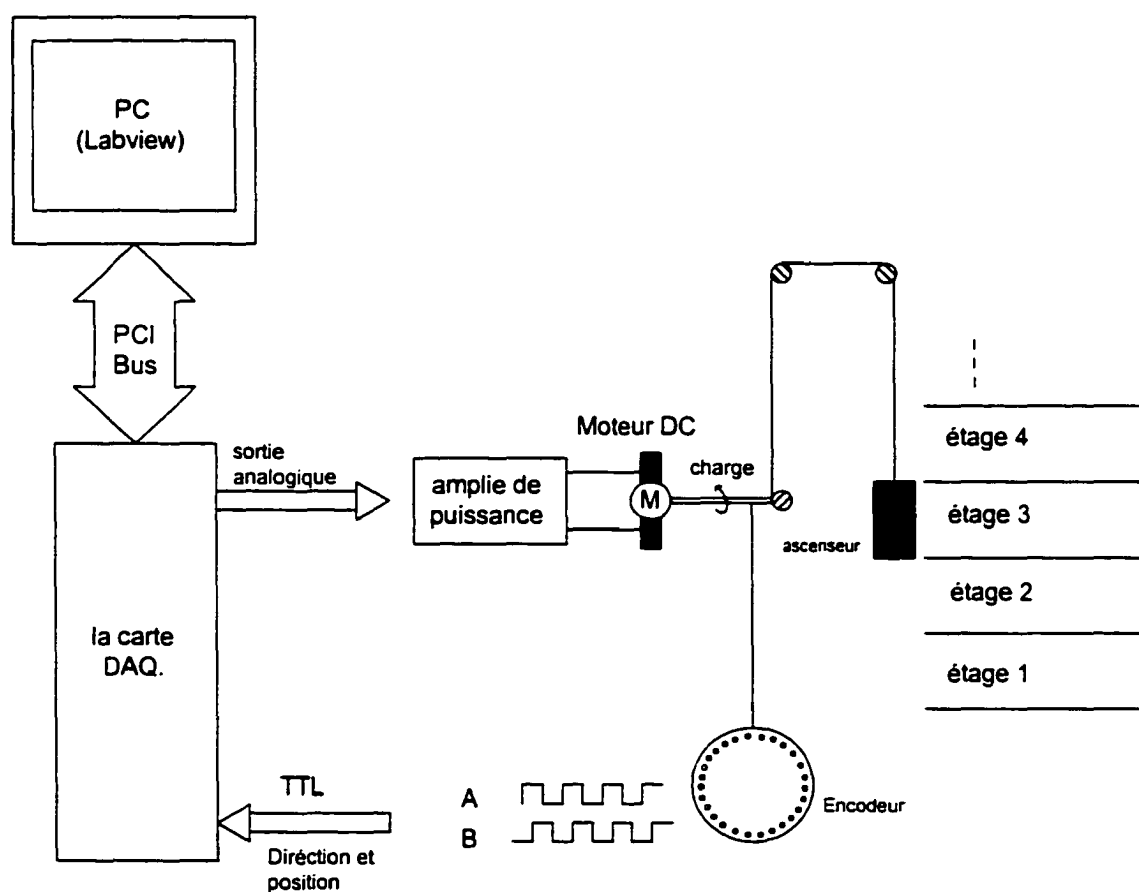


Figure 3. 24 : Le système au complet.

La carte est utilisée pour faire la liaison entre le micro-ordinateur et le moteur qu'on veut asservir en position. Elle peut interpréter les signaux analogiques venant du système puis le commander. Voir Figure 3.25.

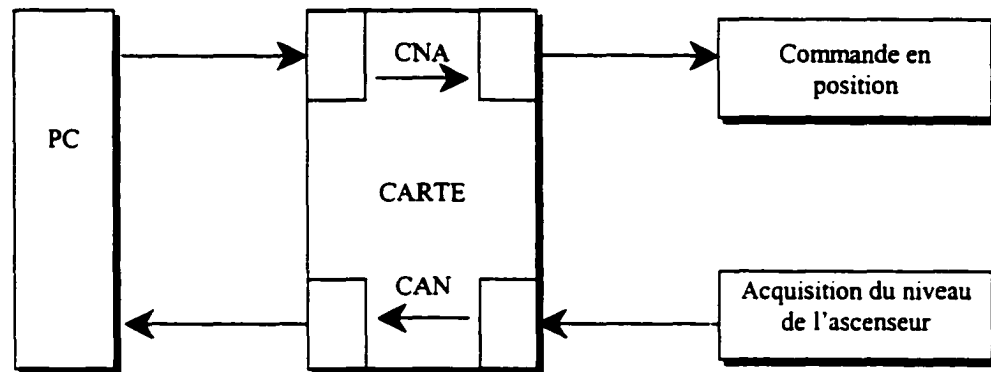


Figure 3. 25 : Interface entre PC et moteur.

La carte d'acquisition DAQ est de type PCI-MIO-16 E-4, elle possède un circuit MITE pour le contrôle du bus sur les cartes PCI (*Peripheral Component Interconnect*), un circuit DAQ-PnP pour la configuration Plug and Play, un circuit DAQ-STC (*System Timing Controller*) pour les opérations de compteurs/timers et un circuit NI-PGIA (*National Instrument- Programmable Gain Instrumentation Amplifier*) pour les mesures d'entrées analogiques de haute précision. Voir le schéma interne de la carte d'acquisition DAQ en annexe D.

Cette carte est de type (plug and play), conçue par "National Instrument" dont le but est l'acquisition de données en temps réel. Elle comporte 16 entrées et 2 sorties analogiques, 8 Entrées/Sorties numériques et deux compteurs (24 bits) qui servent à traiter les signaux TTL.

Les compteurs permettent d'ajouter des fonctions de comptage ou de séquençement de haute précision à notre système d'acquisition de données (DAQ). Ils

transmettent et répondent à des signaux logiques transistor-transistor (TTL), qui sont des signaux à impulsion carrée dont la valeur est 0V (basse) ou 5V (haute). Voir figure 3.26.

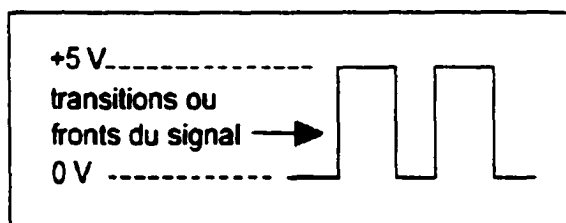


Figure 3. 26 : Un signal TTL

Bien que les compteurs comptent uniquement les transitions (fronts) d'un signal source TTL, on peut utiliser cette capacité de comptage à plusieurs fins :

- On peut générer des impulsions TTL carrées pour créer des signaux d'horloge et de déclenchement pour d'autres applications DAQ.
- On peut mesurer la largeur d'impulsion de signaux TTL.
- On peut mesurer la fréquence et la période de signaux TTL.
- On peut compter les transitions de signaux TTL (fronts) ou mesurer le temps écoulé.
- On peut diviser la fréquence de signaux TTL.

On peut configurer le DAQ-STC pour qu'il puisse compter les transitions montantes ou descendantes de l'entrée SOURCE. Le compteur dispose d'un registre de comptage 24 bits avec une gamme de comptage allant de 0 à 16777215. Le compteur peut être configuré pour une incrémentation ou une décrémentation sur chaque front compté.

La carte DAQ permet d'acquérir et générer des données à partir des E/S analogiques et numériques. Il faut configurer chaque voie qu'on va utiliser. Le logiciel

DAQ de LabVIEW comprend une application de configuration des voies. Pendant la configuration, il faut définir les quantités physiques qu'on va mesurer ou générer sur chaque canal. On peut donner aussi à chaque configuration de voie un nom unique qui sera utilisé pour adresser les voies dans LabVIEW. Les configurations de voies qu'on a définies sont sauvegardées dans un fichier qui indique au driver NI-DAQ comment ajuster et traiter chaque voie DAQ en fonction du nom qui lui a été attribué.

Dans LabVIEW et en particulier dans la palette de fonctions, on peut trouver les VIs (*Virtual Instruments*) pour l'acquisition de données. Il existe six icônes dans une sous-palette qui nous donne accès à six classes différentes de VIs DAQ. Figure 3.27.

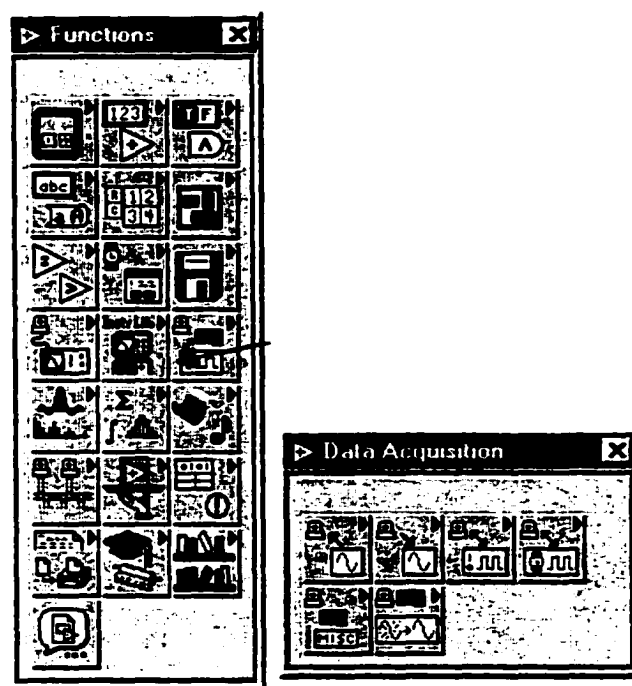


Figure 3. 27 : Sous palettes pour l'acquisition de données.

3.13 Réalisation

Pour réaliser notre projet, on avait besoin de configurer une voie analogique qui nous donne un signal entre -10 et $+10$ V, on va la nommer « DC MOTOR », qui sert à lier la sortie du PID (software) à la puissance du moteur (Hardware), via la carte d'acquisition DAQ qui assure la communication en temps réel, tout en respectant le temps d'échantillonnage.

On a configuré aussi deux autres voies comme entrées analogiques « mesure » et « mesure1 », pour prendre les mesures et vérifier les sorties de notre carte d'acquisition DAQ ainsi que de l'encodeur. Voir Figure 3.28.

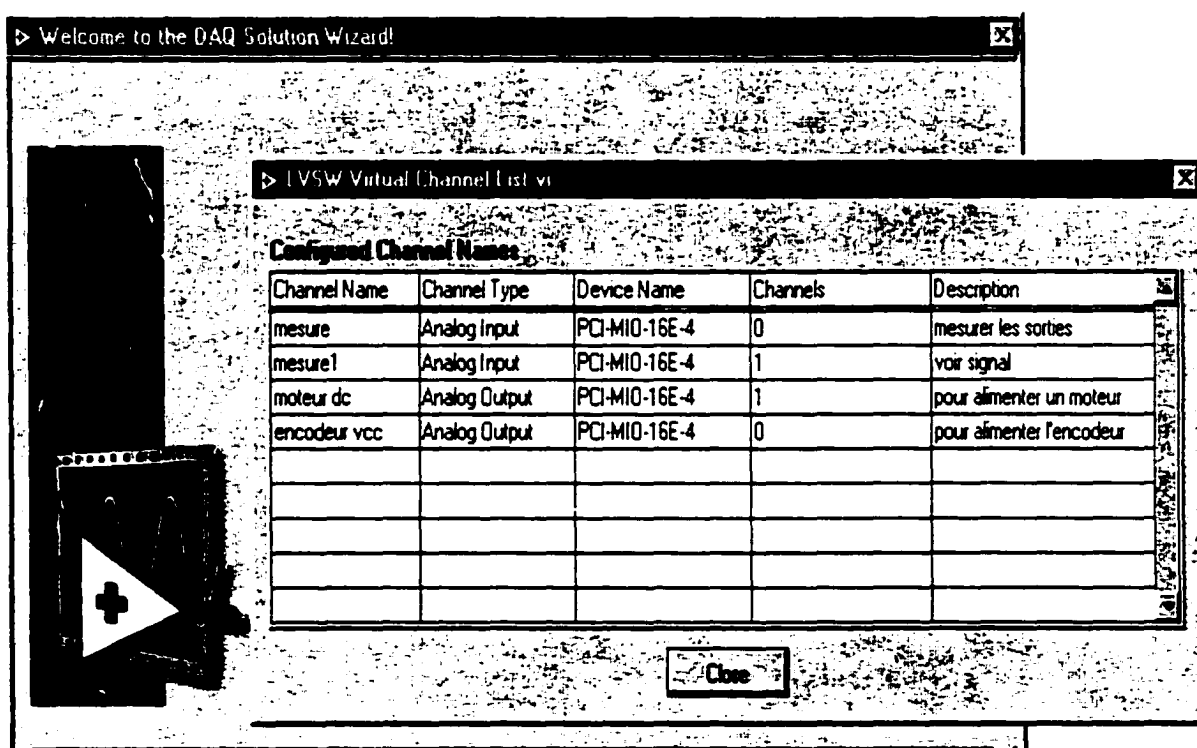


Figure 3. 28 : Voies configurées.

La configuration est faite à l'aide d'un outil disponible avec le logiciel LabVIEW qui est aussi fait par la compagnie "National Instrument". Pour configurer une voie, il faut lui associer un nom unique. Voir figure 3.29. Il faut aussi définir la quantité physique qui entre ou qui sort de cette voie. Toutes ces informations avec les noms seront sauvegardées dans un fichier et utilisées plus tard dans les programmes fait avec LabVIEW. Figure 3.30.

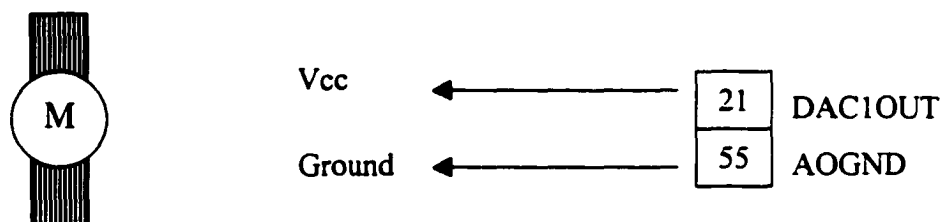


Figure 3. 29 : Voie analogique pour commander le moteur.

Figure 3. 30 : Configuration d'une voie analogique.

Pour vérifier la position du moteur, on a utilisé un compteur pour compter les impulsions qui nous viennent de l'encodeur (Canal A), puis on a réservé une Entrée/Sortie numérique pour examiner l'autre signal de l'encodeur (Canal B).

Ce compteur possède 3 entrées et 2 sorties. Figure 3.31.

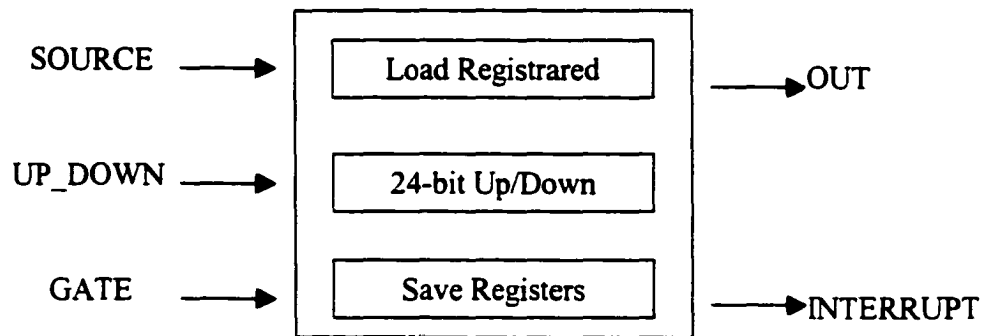


Figure 3. 31 : Schéma simplifié d'un compteur.

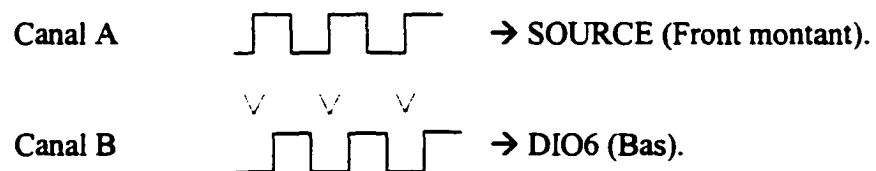
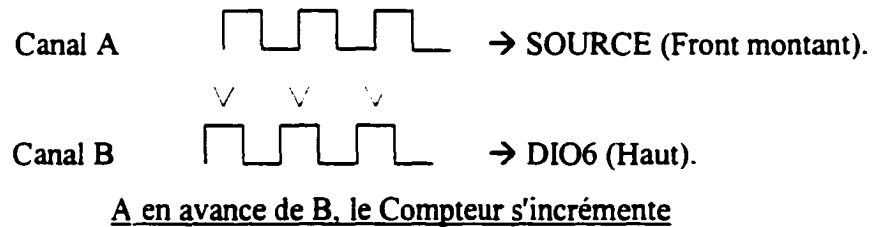


Figure 3. 32 : Compter les impulsions de l'encodeur.

Le canal A de l'encodeur est lié à l'entrée source du compteur, tandis que le canal B est lié à une entrée numérique, pour contrôler l'état (bas/haut). Si cet état est bas, le compteur se décrémente à chaque transition de 0 à 1 du signal lié à la source, par contre si l'état est haut le compteur s'incrémente et compte les impulsions arrivant à la source. Figure 3.32.

Notre encodeur fait 512 impulsions à chaque tour, pour convertir en rad on multiplie par un gain ($k = 2 \pi / 512$). Figure 3.33.

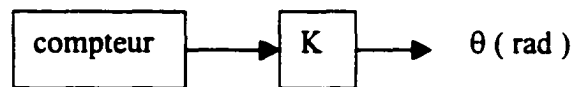


Figure 3. 33 : Lire la position.

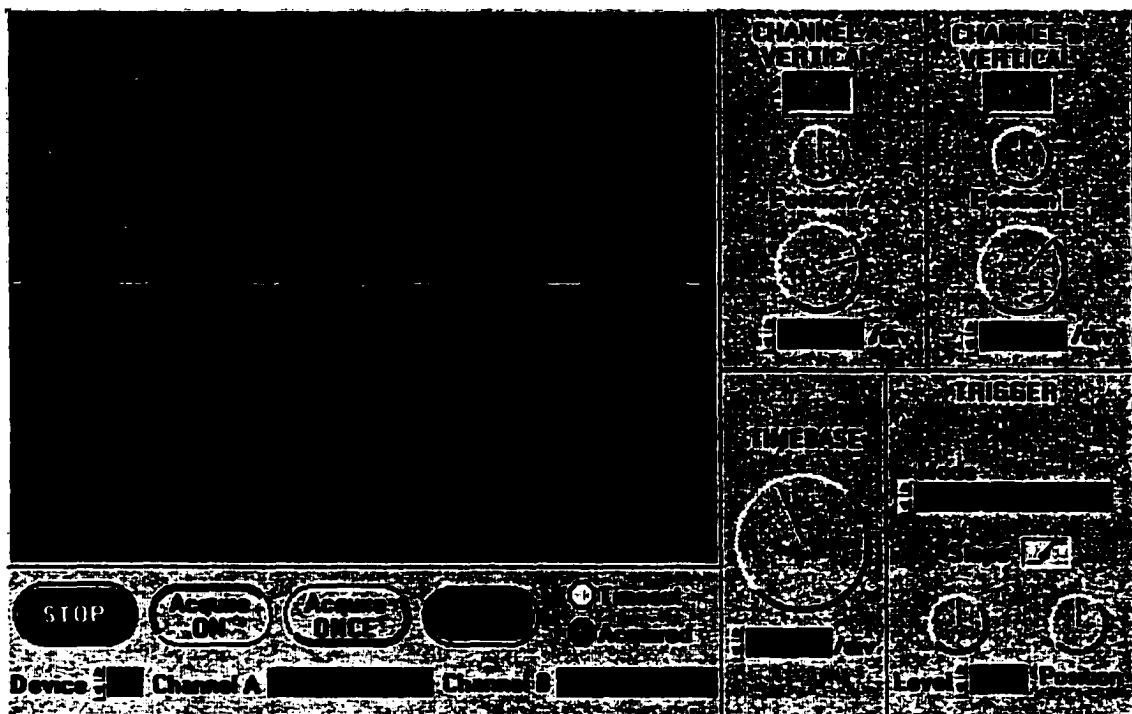


Figure 3. 34 : les signaux A et B de l'encodeur.

On peut voir dans la figure 3.34 les deux signaux A et B, saisis par la carte d'acquisition, (sur calibrage différent), on voit bien le déphasage de 90° entre les deux.

La carte DAQ nous offre deux sorties +5V, on va utiliser une sortie pour alimenter l'encodeur, le schéma de connexions sera la suivante :

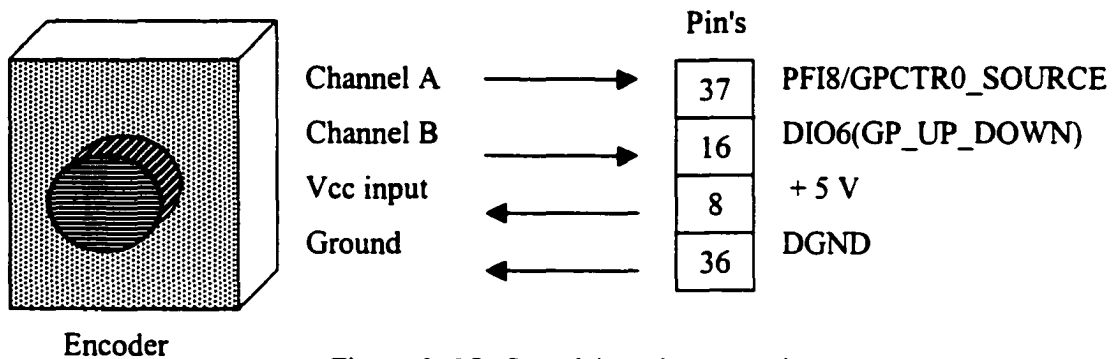


Figure 3. 35 : Le schéma de connexions.

Pour le contrôle, on va utiliser un contrôleur PID qui a été fait avec LabVIEW, le schéma de contrôle sera le suivant : Figure 3.36.

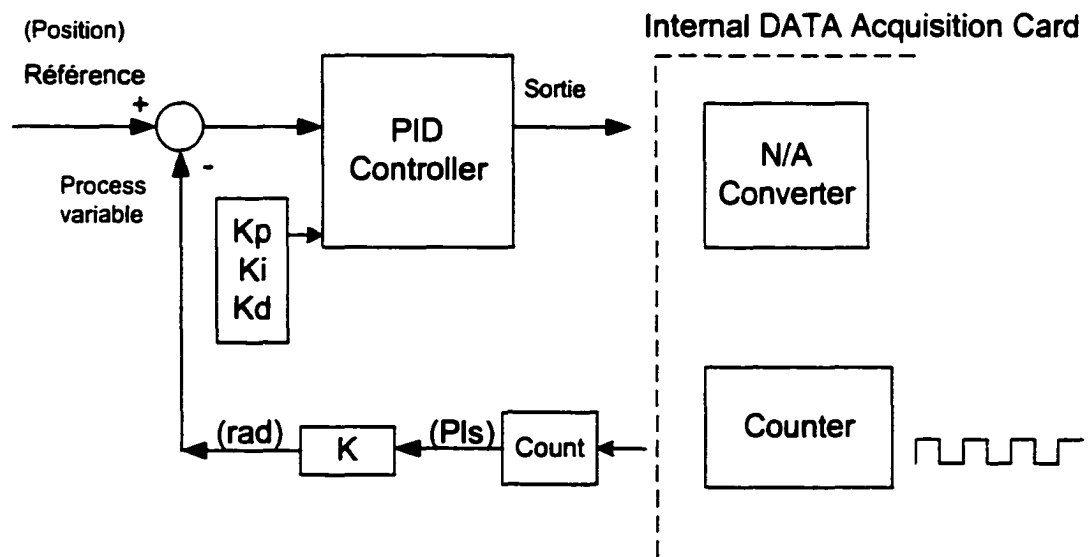


Figure 3. 36 : Contrôle avec PID.

3.14 Identification des paramètres

Il existe un très grand nombre de méthodes d'identification. Parmi lesquelles on va choisir celle de la boucle ouverte.

Cette méthode consiste à envoyer un signal test à l'entrée du procédé qu'on veut caractériser et à enregistrer la sortie produite. Comme on applique directement le signal test sur l'entrée du système, il faut alors que la sortie soit exploitable et ne diverge pas, donc pour cette méthode il faut que le système soit stable. Mais il est possible d'identifier des systèmes intrinsèquement instables mais il faut d'abord les boucler afin que le système à identifier soit stable. On dit alors qu'on procède à une identification en boucle fermée.

Pour identifier notre moteur, on a fait un programme avec LabVIEW. Ce programme nous donne directement les paramètres du moteur et il peut être appliqué pour identifier les paramètres de n'importe quel autre moteur. Voir figure 3.37.

L'utilisateur peut effectuer plusieurs expériences en faisant varier la tension appliquée au moteur (Set point) et le temps d'échantillonnage (Sampling time). L'application permet d'afficher la position du moteur en fonction du temps, ainsi que la vitesse. Alors l'utilisateur peut étudier la réponse du système, quantifier le temps de réponse, la valeur finale de la vitesse, la constante du temps τ et le gain statique K . Il peut en tout temps imprimer les graphes obtenus et observer les nombres des impulsions générés par l'encodeur. À la fin toutes les informations (Position, Vitesse et le temps de chaque échantillon) seront sauvegardées sous forme des vecteurs dans un fichier texte pour des consultations ultérieures.

Pour identifier les paramètres, on a appliqué une tension de 10 volts aux bornes du moteur jusqu'à l'établissement du régime permanent. Puis on a prélevé, en boucle ouverte, le gain statique K et la constante du temps τ .

Ce gain est calculé à partir de la vitesse maximum du moteur et il est donné par :

$$K = V_{\max}/V \quad (3.27)$$

Avec :

V_{\max} : Vitesse maximale en (rad/sec).

V = Tension appliquée en (volt).

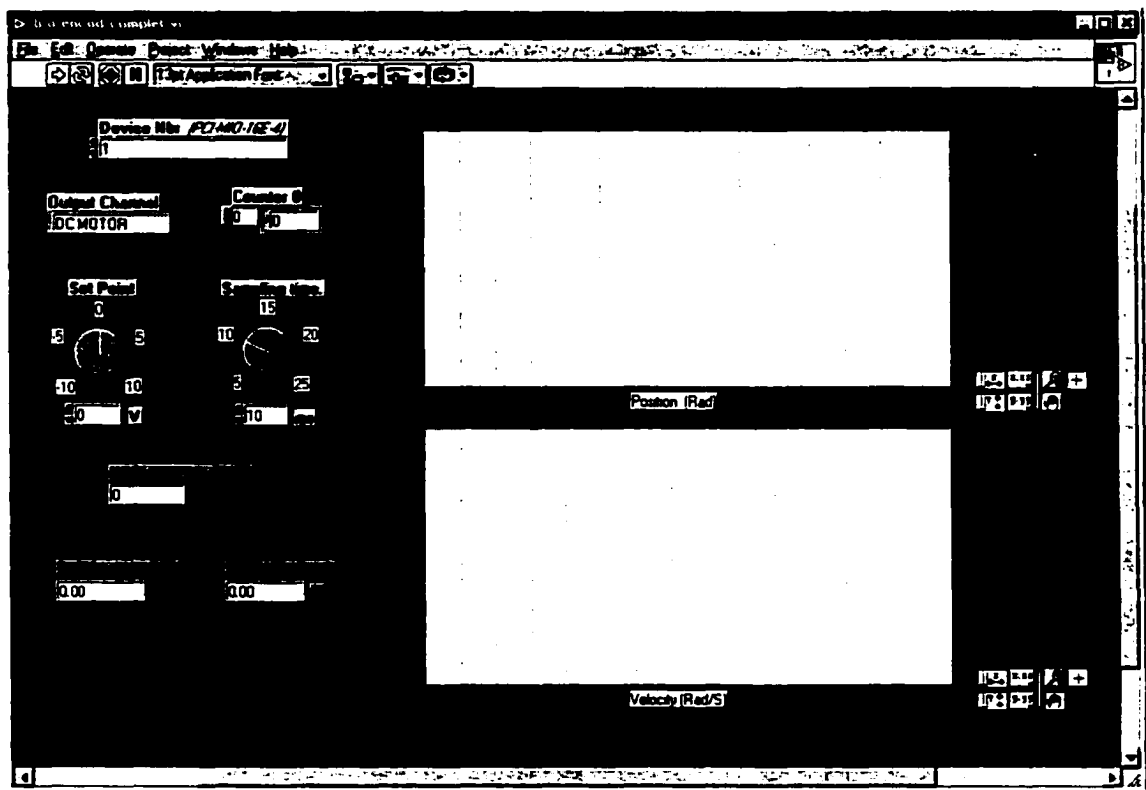


Figure 3. 37 : Interface d'identification.

Application numérique :

$$k = \frac{170}{5} = 34$$

Pour trouver la constante de temps τ , on a cherché la vitesse qui correspond à 63% du régime permanent, alors τ est le temps où le moteur atteint cette valeur.

La valeur qu'on a trouvée pour τ est 50ms.

D'autre part pour une fonction de transfert $G(s) = \frac{K}{s^2\tau + s}$

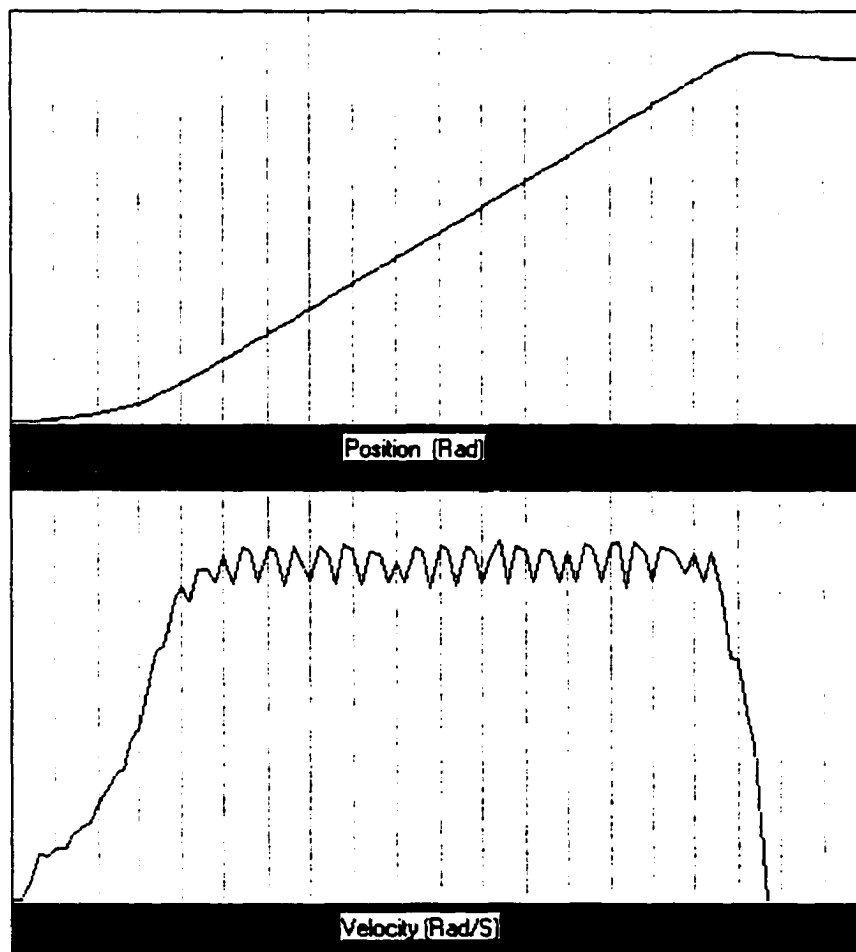


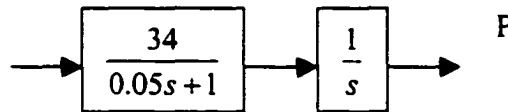
Figure 3. 38 : Position et vitesse du moteur.

3.15 Simulation

On va utiliser les paramètres trouvés pour simuler le comportement de notre système afin de vérifier les résultats qu'on va avoir dans la pratique. La simulation est faite avec Matlab et avec LabVIEW.

3.15.1 En boucle ouverte

La fonction du système en boucle ouverte est la suivante :



On peut déterminer la vitesse et la position correspondantes à la sortie de notre moteur. Voir les programmes sous Matlab en annexe E.

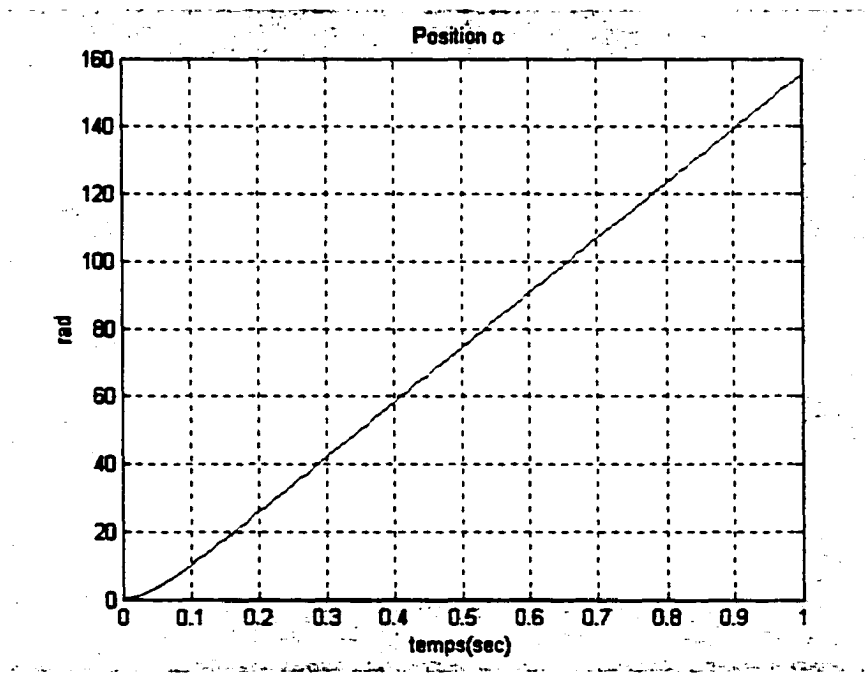


Figure 3. 39 : Position en boucle ouverte (Matlab).

Cette sortie est celle de la réponse à un échelon, elle représente la position (Figure 3.39) et la vitesse (Figure 3.40).

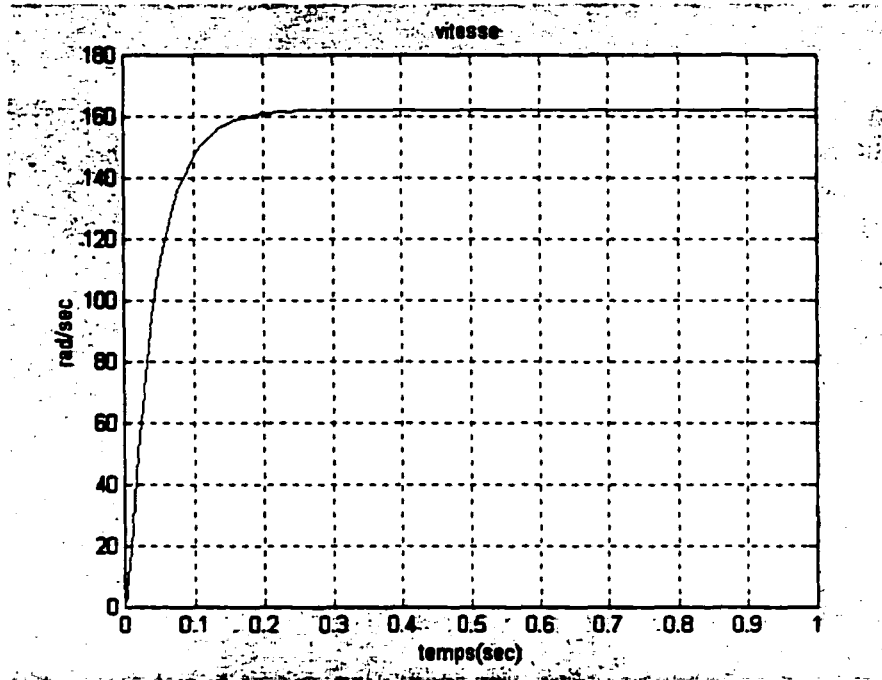


Figure 3. 40 : Vitesse en boucle ouverte (Matlab).

3.15.2 En boucle fermée

La simulation en boucle fermée avec Matlab sans utiliser un régulateur nous donne les résultats suivants :

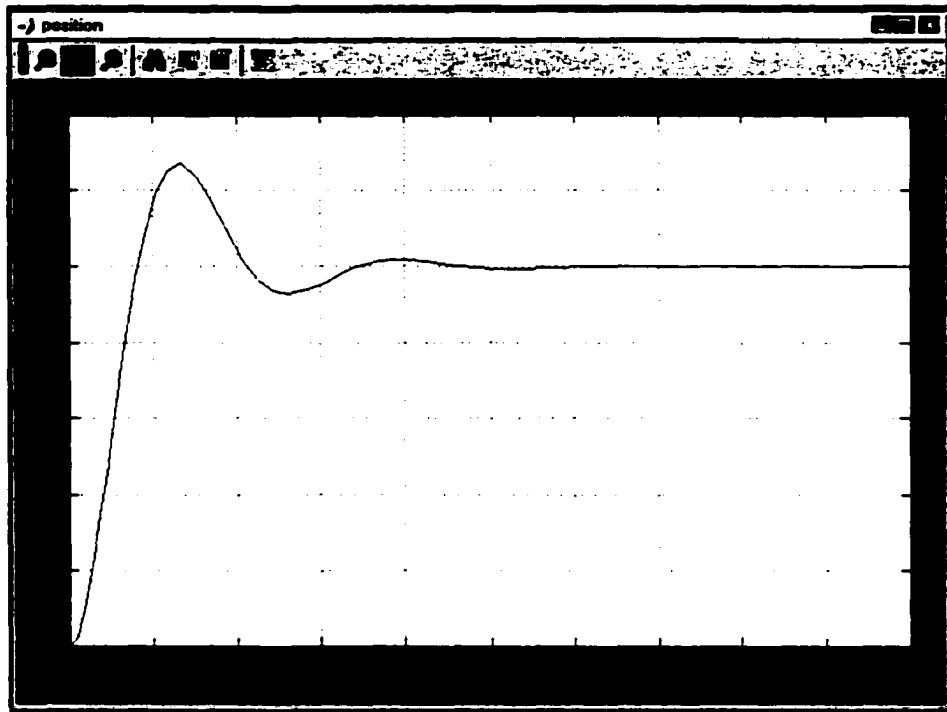


Figure 3. 41 : Position en boucle fermée sans PD (Simulink).

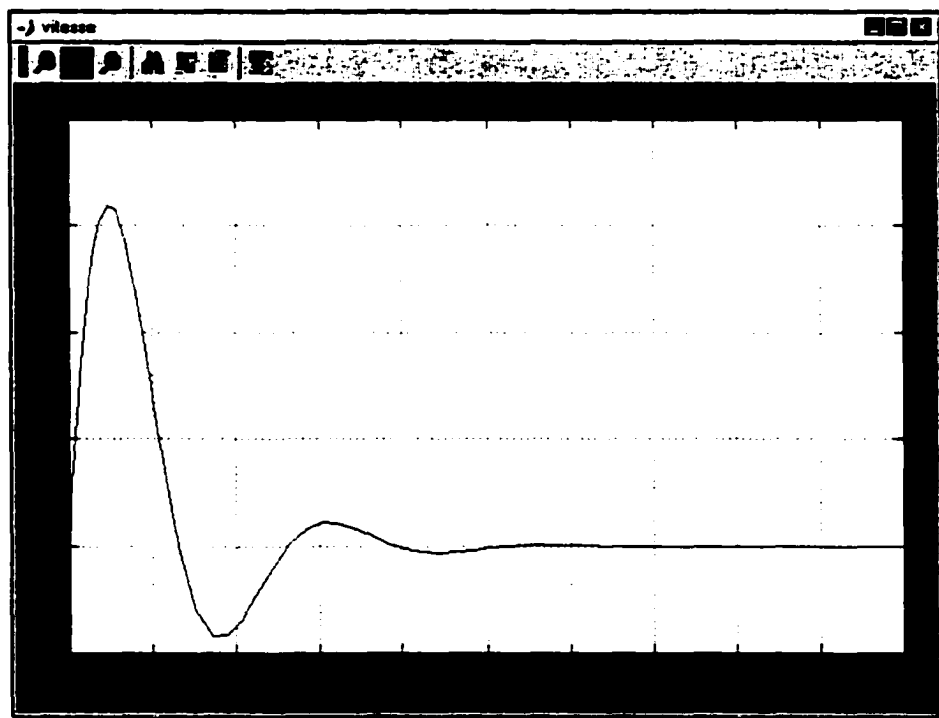


Figure 3. 42 : Vitesse en boucle fermée sans PD (Simulink).

3.15.3 Conception du P.D.

A l'aide de la réponse en fréquence du système en boucle ouverte produite par les diagrammes d'amplitude et de phase, on peut observer quels paramètres doivent être modifiés pour avoir un système stable et précis en ajoutant un régulateur P.D (proportionnel-dérivée).

Le schéma simplifié de ce régulateur est représenté par la figure 3.43.

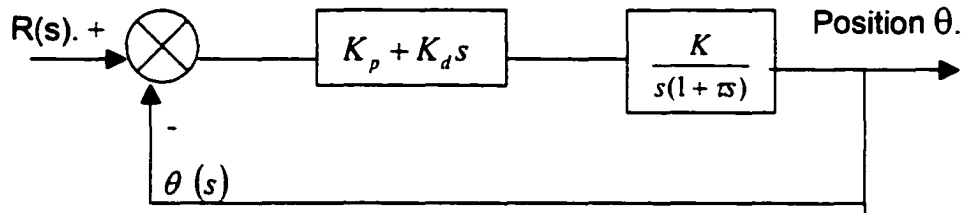


Figure 3. 43 : Commande de la position par régulateur PD.

$$G(s) = \frac{K}{s(1 + \tau s)} \quad (3.28)$$

$$P(s) = K_p + K_d s \quad (3.29)$$

En boucle fermée on a :

$$\frac{\theta(s)}{R(s)} = \frac{P(s)G(s)}{1 + P(s)G(s)} \quad (3.30)$$

$$P(s)G(s) = (K_p + K_d s) * \frac{K}{s(s\tau + 1)} = \frac{K(K_d s + K_p)}{s(s\tau + 1)} \quad (3.31)$$

$$\frac{\theta(s)}{R(s)} = \frac{K(K_d s + K_p)}{s(s\tau + 1) + K(K_d s + K_p)} \quad (3.32)$$

$$= \frac{K/\tau (K_d s + K_p)}{s^2 + s \left(\frac{KK_d + 1}{\tau} \right) + \frac{KK_p}{\tau}} \quad (3.33)$$

Cette fonction de transfert a une équation caractéristique du 2^{ème} ordre qui nous aidera à définir le comportement optimal du système par l'obtention des racines et des valeurs correspondantes de K_p et K_d .

La conception du P.D. sera faite par la méthode de placement des pôles :

Calculs de ζ (facteur d'amortissement) et de ω_n (fréquence de coupure)

Hypothèses :

$T_r = 2$ sec. (temps de réponse du système)

Dépassement de 5%

Calcul de ω_n

$$T_r = -\frac{1}{\zeta \omega_n} \ln \left[0.05 \sqrt{1 - \zeta^2} \right] \quad (3.34)$$

$$\omega_n = 2.405 \text{ rad/s}$$

Calcul de ζ

$$0.05 = e^{\left(\frac{-\pi}{\sqrt{1 - \zeta^2}} \right)} \quad (3.35)$$

$$\zeta = 0.69$$

Équation caractéristique du système

$$\text{Soit : } D(s) = s^2 + s \left(\frac{KK_d + 1}{\tau} \right) + \frac{KK_p}{\tau} = s^2 + a_1 s + a_0 \quad (3.36)$$

$$\text{On peut aussi écrire : } D(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2) \quad (3.37)$$

La partie réelle des pôles conjugués vaut $\zeta\omega_n = 1.659$.

Alors :

$$D(s) = (s^2 + 3.332s + 5.6644) = s^2 + 3.318s + 5.784$$

Avec $K=34$ et $\tau=0.05$ on calcule les valeurs de K_d et K_p :

$$K_p = 1.3662$$

$$K_d = 0.0169$$

La vérification avec Matlab, figure 3.44, nous donne ces résultats :

$$\omega_n = 2.4050$$

$$K_p = 1.3662$$

$$K_d = 0.0169$$

```
K=34; % paramètres du moteur
tau=.05;
d=0.15;
tr=.2;
delta=0.05;
ksi=sqrt(log(d)^2/(pi^2+log(d)^2));
wn=-log(delta*sqrt(1-ksi^2))/(ksi*tr)
Kp=wn^2*tau/K
Kd=(2*ksi*wn*tau-1)/K
```

Figure 3. 44 : Vérification avec Matlab.

En utilisant les paramètres trouvés du PD, on obtient le graphe suivant :

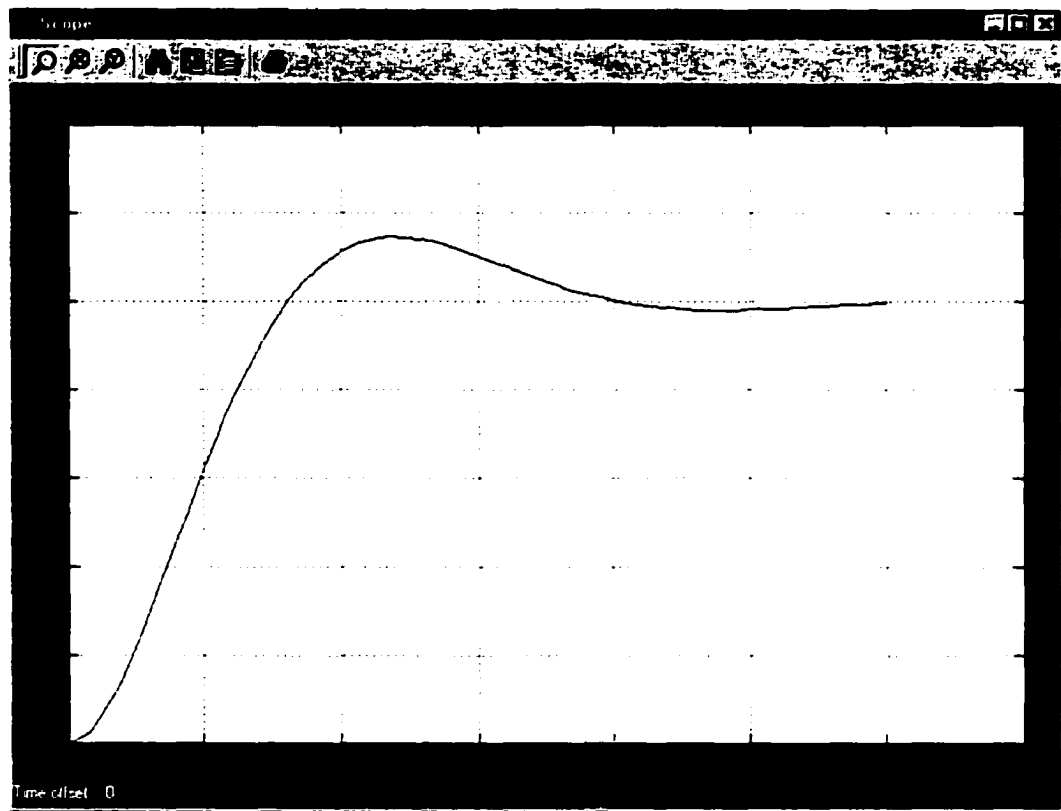


Figure 3. 45 : Position en boucle fermée avec un régulateur PD.

En utilisant les résultats obtenus dans la simulation en boucle fermée sans PD, on constate que notre système atteint son régime permanent après un certain temps, néanmoins on a un dépassement au début, à cause que c'est un système de second ordre.

En utilisant un régulateur PD dans notre simulation, on remarque que le dépassement est atténué, tandis que le temps de réponse augmente à cause de la présence du dérivateur.

3.16 Résultat avec LabVIEW

Après avoir identifier les paramètres du moteur, le gain statique K et la constante du temps τ , on a fait un programme avec LabVIEW, figure 3.46, pour calculer les paramètres d'un régulateur PID, on a utilisé la méthode de placement des pôles.

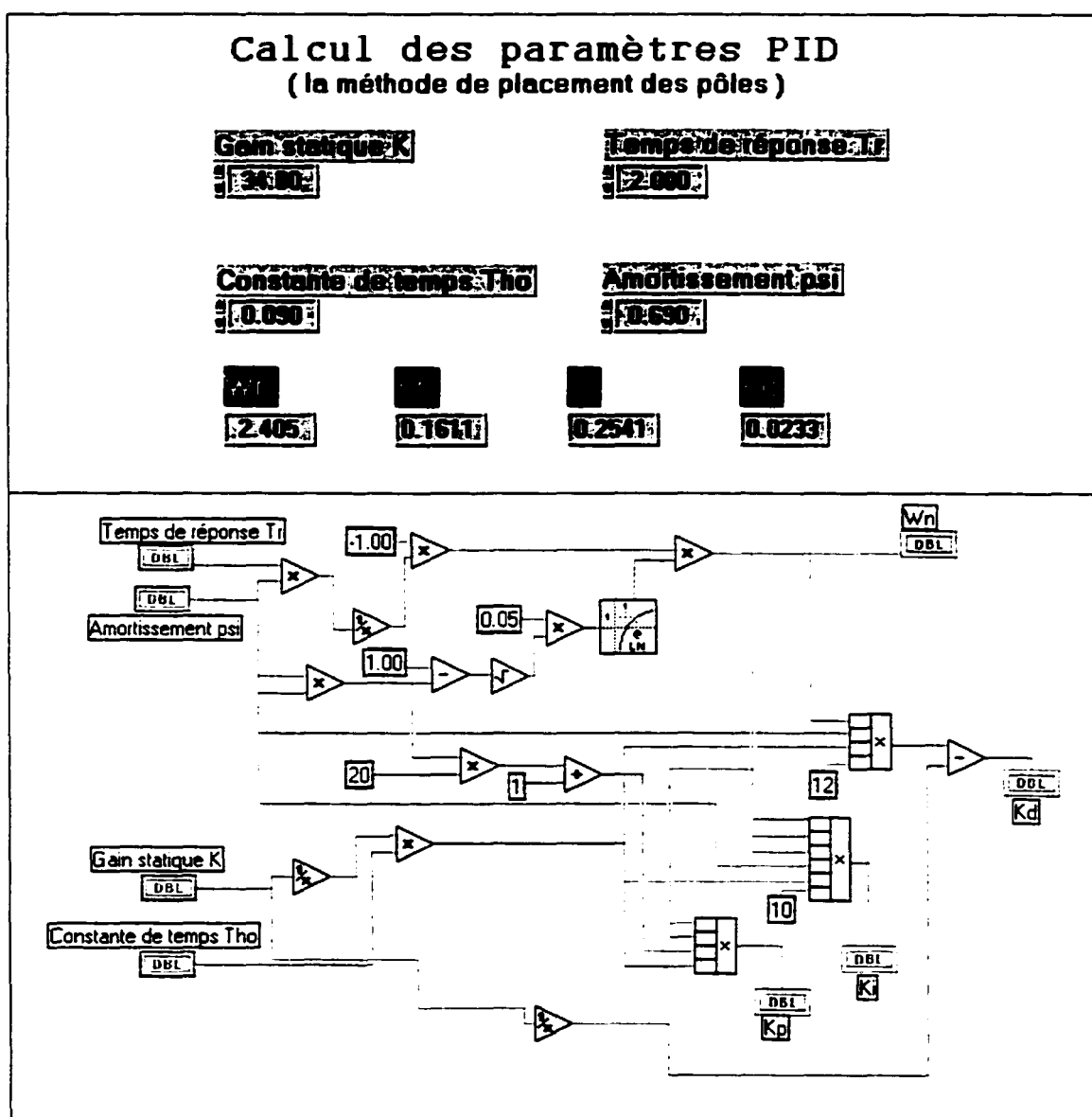


Figure 3. 46 : Calcul du PID.

Ce régulateur sera utilisé par la suite dans l'application qui va contrôler le moteur. Les résultats trouvés sont identiques à ceux obtenus avec Matlab.

Pour simuler le comportement du moteur, on a fait une autre application avec LabVIEW. Dans cette application on peut fixer les paramètres du moteur, les paramètres d'un régulateur PD, le temps d'échantillonnage et le temps de simulation. Voir figure 3.47. Elle nous permet de simuler n'importe quel moteur, il suffit de changer les paramètres K et T_m , comme elle permet aussi de vérifier les résultats qu'on va avoir par la suite quand on va contrôler le moteur en temps réel.

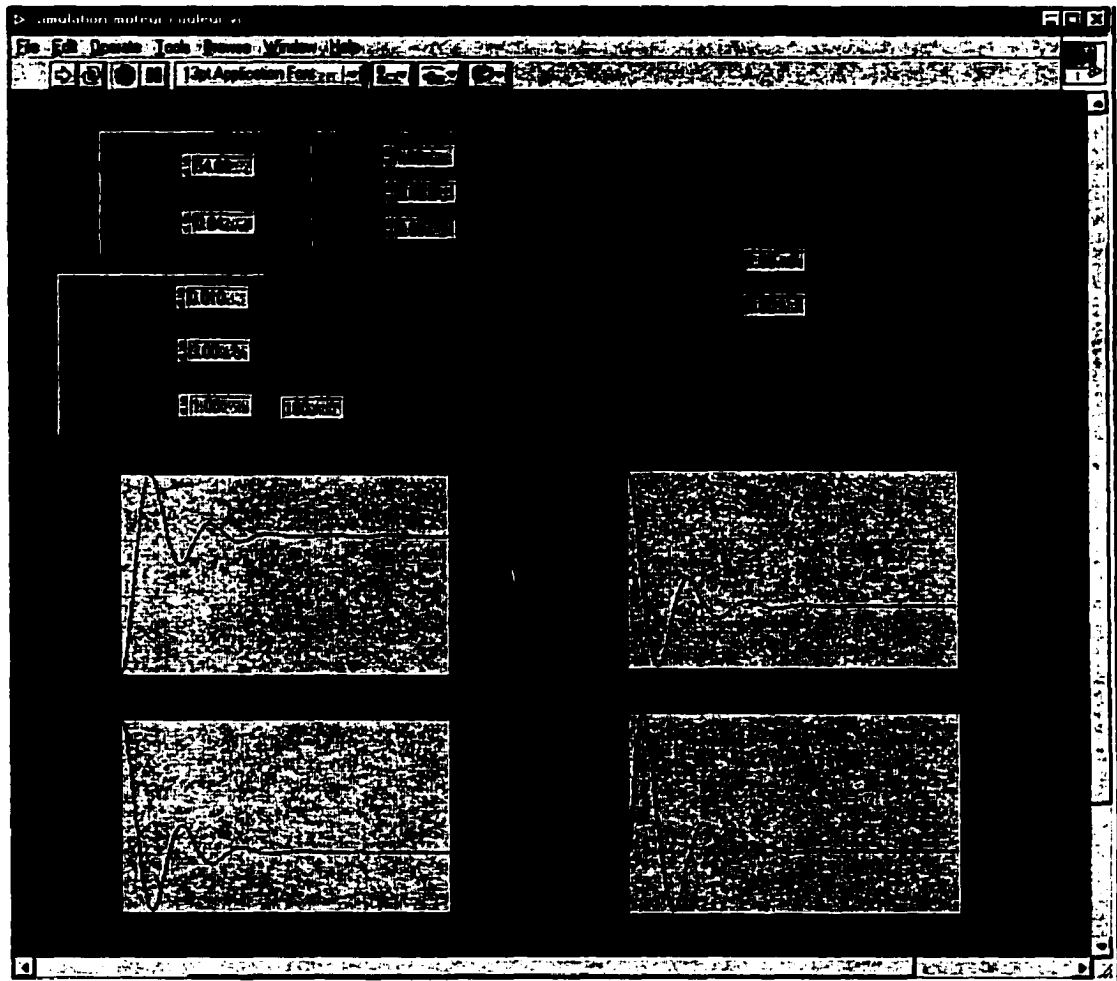


Figure 3. 47 : La simulation avec LabVIEW.

À la fin, on a fait l'application qui nous permet de contrôler le moteur en temps réel. Cette interface nous donne la possibilité de bien manipuler le moteur. On peut varier les paramètres du PID et choisir le temps d'échantillonnage tout en respectant l'acquisition en temps réel. Avec cette application, l'utilisateur peut demander une position quelconque et le système répond en suivant la consigne demandée et en affichant cette consigne et la position actuelle en temps réel. Voir figure 3.48.

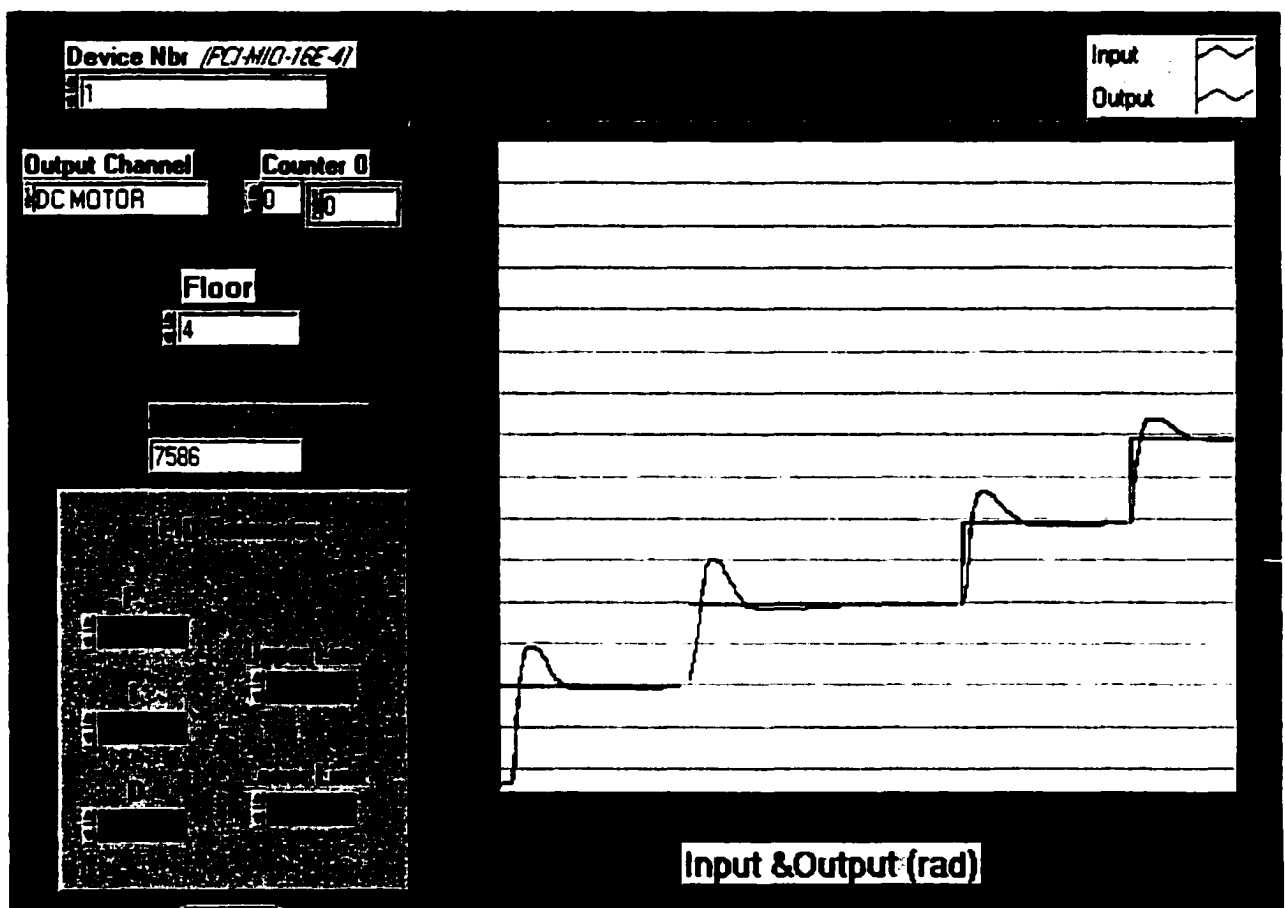


Figure 3. 48 : L'interface du contrôle en temps réel.

Pendant un temps d'échantillonnage T_n , notre application va effectuer les tâches suivantes :

1. Lire la position actuelle du moteur $P_v(n)$
2. Calculer l'erreur (entrée du PID), qui représente la différence entre la position actuelle et la position demandée : $E(n) = S_p(n) - P_v(n)$
3. Calculer la commande (sortie du PID), suivant les paramètres du PID.
4. Appliquer la commande vers le moteur via la carte d'acquisition
5. Attendre le reste du temps pour commencer un autre temps d'échantillonnage $T(n+1)$

Ce schéma représente les tâches effectuées dans chaque temps d'échantillonnage $T(n)$.

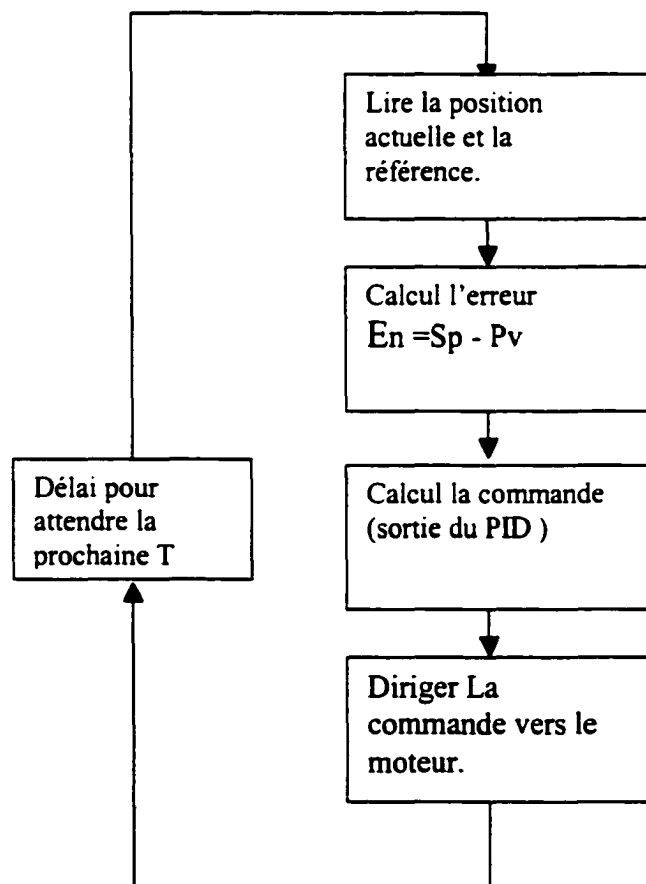


Figure 3. 49 : Algorithme suivi pour contrôler le moteur en temps réel.

Les graphes suivants montrent les résultats obtenus en contrôlant le moteur en temps réel, le premier graphe est obtenu avant l'utilisation d'un régulateur PID, d'où on peut voir la nécessité d'un tel régulateur. Dans le deuxième on peut voir l'amélioration. L'erreur a beaucoup diminué et les résultats sont bien meilleurs que les premiers.

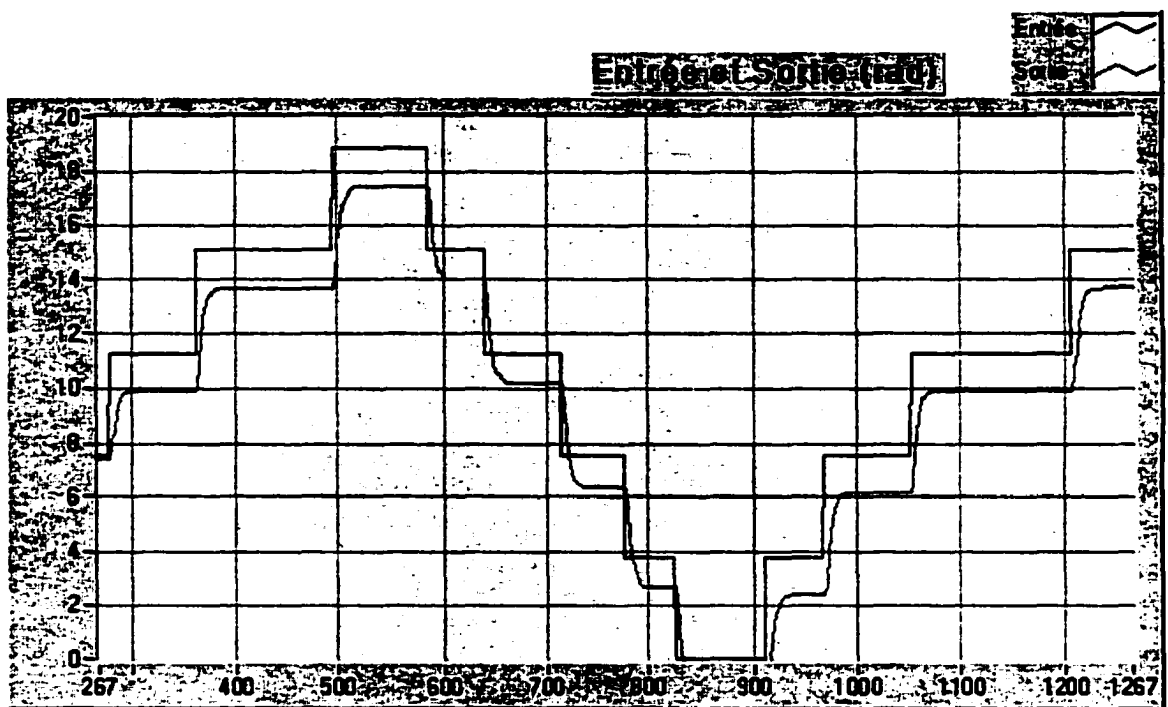


Figure 3. 50 : Résultats sans PID.

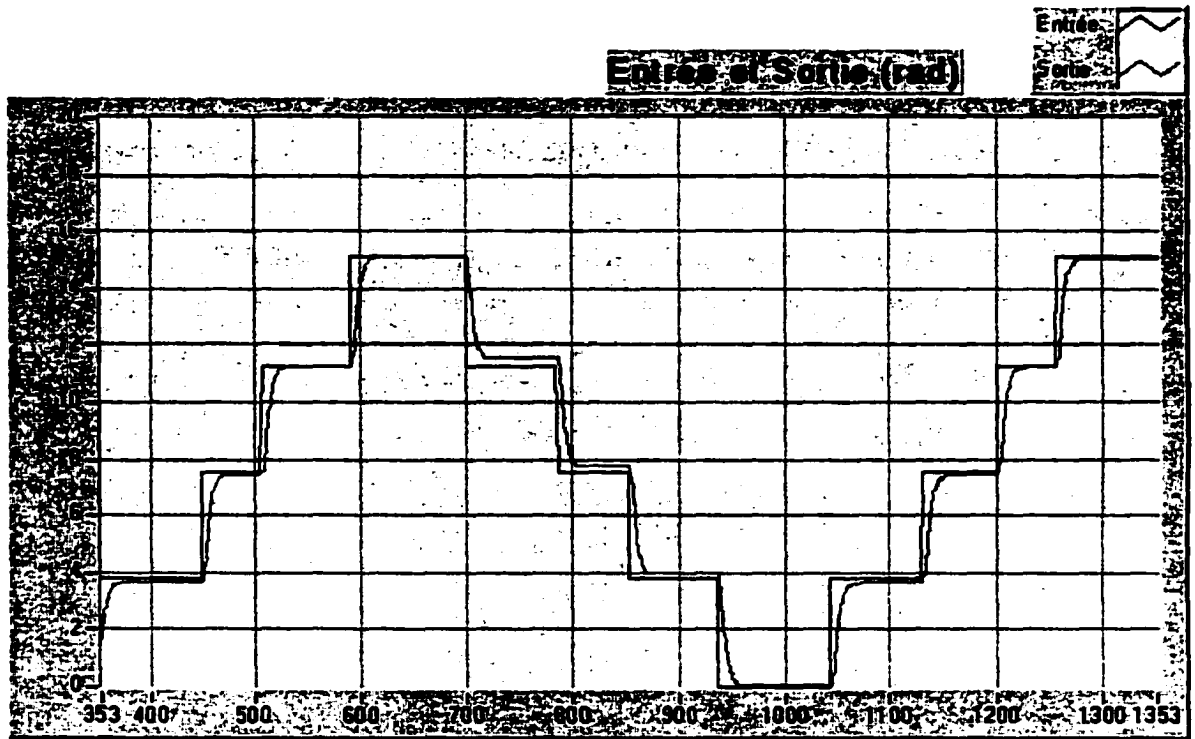


Figure 3. 51 : Résultats avec PID.

3.17 Le contrôle à distance en utilisant le protocole TCP/IP

Pour le contrôle à distance, on a utilisé les protocoles TCP/IP. On peut trouver la suite de ces protocoles dans LabVIEW, sa palette de fonctions possède un ensemble de fonctions TCP/IP, comme illustré dans la figure 3.52, qu'on peut utiliser pour créer des applications clients ou serveurs.

Chaque hôte d'un réseau IP possède une adresse Internet unique de 32 bits. Cette adresse identifie le réseau auquel l'hôte est relié sur Internet, ainsi que l'ordinateur

spécifique sur ce réseau. On peut utiliser cette adresse pour identifier l'expéditeur ou le destinataire des données.

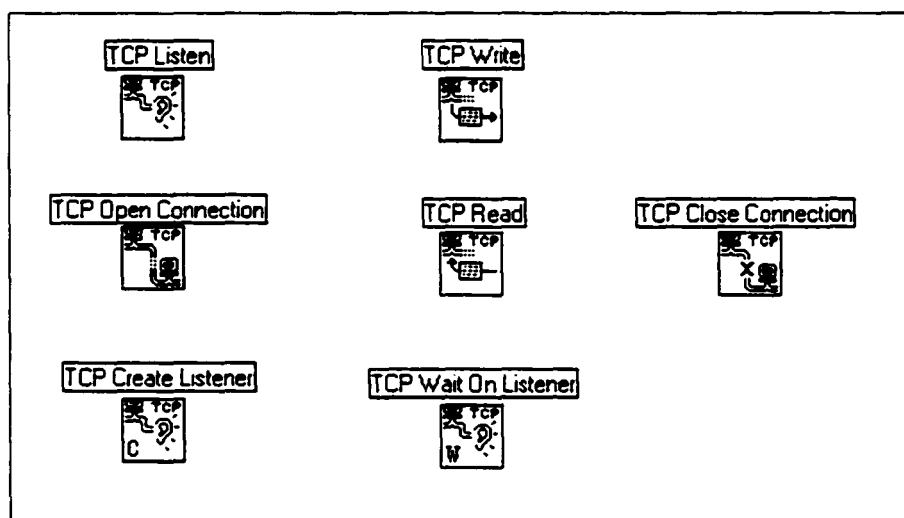


Figure 3. 52 : Les fonctions TCP/IP avec LabVIEW.

TCP est un protocole basé sur des connexions, ce qui signifie que les sites doivent établir une connexion avant de transférer des données. Il permet plusieurs connexions simultanées.

On peut initier une connexion en attendant une connexion en amont ou en recherchant activement une connexion avec une adresse spécifiée. En établissant les connexions TCP, on doit préciser à la fois l'adresse et un port à cette adresse. Un port est représenté par un nombre entre 0 et 65535. Différents ports à une adresse donnée identifient différents services à cette adresse et facilitent la gestion de plusieurs connexions simultanées.

On peut établir activement une connexion avec une adresse et un port spécifiques grâce à la fonction « TCP Open Connection ». Cette fonction nous permet de préciser l'adresse et le port avec lequel on souhaite communiquer. Si la connexion a réussi, la

fonction retourne une identification (ID) de connexion qui identifie cette connexion de façon unique. On utilise cette ID de connexion pour se référer à la connexion lors d'appels de fonctions ultérieures.

Il existe deux méthodes pour attendre une connexion en amont :

- Première méthode : On utilise la fonction « TCP Listen » pour créer un auditeur et puis on attend une connexion TCP acceptée à un port spécifié. Si la connexion est réussie, la fonction retourne une ID de connexion, ainsi que l'adresse et le port TCP à distance.

- Seconde méthode : On utilise la fonction « TCP Create Listener » pour créer un auditeur, puis on utilise la fonction « TCP Wait On Listener » pour déceler et accepter de nouvelles connexions. La fonction « TCP Wait On Listener » retourne la même ID d'auditeur que celle transmise à la fonction, ainsi que l'ID de connexion. Lorsqu'on n'attend plus de nouvelles connexions, on peut utiliser la fonction « TCP Close Connection » pour fermer un auditeur. Dans ce cas on ne peut ni lire ni écrire dans un auditeur.

La seconde méthode est avantageuse, car elle permet d'annuler une opération d'écoute en appelant la fonction « TCP Close Connection ». Ceci est utile dans le cas où on souhaite écouter une connexion sans utiliser de time-out (délai d'attente), mais où on souhaite annuler l'écoute lorsqu'une certaine autre condition devient vraie (par exemple, lorsque l'utilisateur appuie sur un bouton).

Lorsqu'une connexion est établie, on peut lire et écrire des données dans l'application à distance grâce aux fonctions « TCP Read » et « TCP Write ».

Enfin, on utilise la fonction « TCP Close Connection » pour fermer la connexion de l'application à distance. S'il y a des données non lues et si la connexion se ferme, ces données risquent d'être perdues. Les parties connectées doivent utiliser un protocole de niveau supérieur pour déterminer le moment où fermer la connexion. Une fois qu'une connexion est fermée, on ne peut ni relire ni réécrire dessus.

3.17.1 Algorithme d'une application client - serveur

On trouve ci-après une description généralisée pour créer une application client serveur :

1) Cette partie explique l'utilisation des composantes du modèle de diagramme client avec TCP.

On utilise la fonction « TCP Open Connection » pour ouvrir une connexion sur un serveur. On doit préciser l'adresse Internet du serveur (IP), ainsi que le port du serveur. L'adresse identifie un ordinateur sur le réseau. Le port est un numéro supplémentaire qui identifie un canal de communication sur l'ordinateur utilisé par le serveur pour écouter des requêtes de communication. Lorsqu'on crée une application TCP serveur, on doit préciser le port que le serveur utilisera pour la communication.

Pour exécuter une commande sur le serveur, on utilise la fonction « TCP Write » pour envoyer la commande au serveur. On utilise ensuite la fonction « TCP Read » pour relire les résultats. Avec la fonction « TCP Read », on doit préciser le nombre de caractères qu'on souhaite lire. Comme la longueur de la réponse peut varier, ceci peut s'avérer peu commode. Le serveur peut présenter le même problème avec la commande, car la longueur d'une commande peut aussi varier.

On peut utiliser les méthodes suivantes pour adresser des commandes de tailles différentes :

- Faire précéder la commande et le résultat d'un paramètre de taille fixe qui spécifie la taille de la commande ou du résultat. Dans ce cas, on lit le paramètre de taille, puis on lit le nombre de caractères précisé par la taille. Cette option constitue une méthode efficace et souple.

- Donner une taille fixe à chaque commande et résultat. Lorsque la taille d'une commande est inférieure à la taille précisée, on peut l'augmenter jusqu'à obtenir la taille spécifiée.

- Faire suivre chaque commande et résultat d'un caractère de terminaison spécifique. On doit alors lire les données par petites sections jusqu'à ce qu'on obtienne le caractère de terminaison.

À la fin on utilise la fonction « TCP Close Connection » pour fermer la connexion au serveur.

2) Dans cette section on explique comment on peut utiliser TCP pour exécuter chaque composante du modèle de serveur général.

Comme aucune initialisation n'est nécessaire avec TCP, on peut ignorer cette étape.

On utilise la fonction « TCP Listen » pour attendre une connexion. On doit préciser le port qui sera utilisé pour la communication. Ce port doit être le même que celui auquel le client essaie de se connecter.

Si une connexion est établie, on effectue une lecture à partir de ce port pour récupérer une commande. Comme indiqué dans l'exemple du client TCP, on doit décider le format des commandes. Si les commandes sont précédées d'un champ de longueur, on lit d'abord ce champ puis on lit la quantité de données indiquée par le champ.

L'exécution d'une commande doit être indépendante du protocole car elle est réalisée sur l'ordinateur local. Lorsque l'exécution est terminée, on transmet les résultats à la prochaine étape, où ils sont transmis au client.

On utilise la fonction « TCP Write » pour retourner des résultats. Les données doivent se trouver dans un format accepté par le client.

Le protocole TCP gère aisément plusieurs connexions. On peut utiliser les méthodes décrites dans la section précédente pour implanter les composantes d'un serveur avec plusieurs connexions.

3.17.2 Time-Out et erreurs

Lorsqu'on conçoit une application réseau, on doit considérer avec attention ce qui se passera en cas de problème. Par exemple, si le serveur tombe en panne, comment ce problème sera-t-il géré par chaque client ?

Une solution est de s'assurer que chaque client possède un time-out (délai d'attente). Si un résultat attendu ne se produit pas, le client va continuer l'exécution pendant un certain temps. En continuant, le client peut essayer de rétablir l'exécution ou de rapporter l'erreur. Si nécessaire, il peut arrêter l'application client.

On doit mentionner que c'est le client qui doit indiquer l'adresse IP du serveur et non pas l'inverse. Si on travaille sur la même machine, on peut écrire « localhost » au lieu d'écrire l'adresse de cette machine. Il faut mentionner aussi qu'il faut lancer toujours l'application serveur avant de lancer l'application client, parce que le client envoie un appel de connexion et le serveur doit être à l'écoute pour répondre à la demande. Pour chaque connexion, il doit avoir un port. Si on veut envoyer et recevoir des données dans une même application, alors on doit avoir deux connections et un port pour chacune d'elles. On peut choisir un numéro quelconque du port (entre 0 et 65535), à condition qu'il ne soit pas utilisé par une autre application.

3.17.3 Applications réalisées

Avant de commencer notre application, on a fait quelques programmes pour comprendre mieux le fonctionnement du TCP/IP avec LabVIEW.

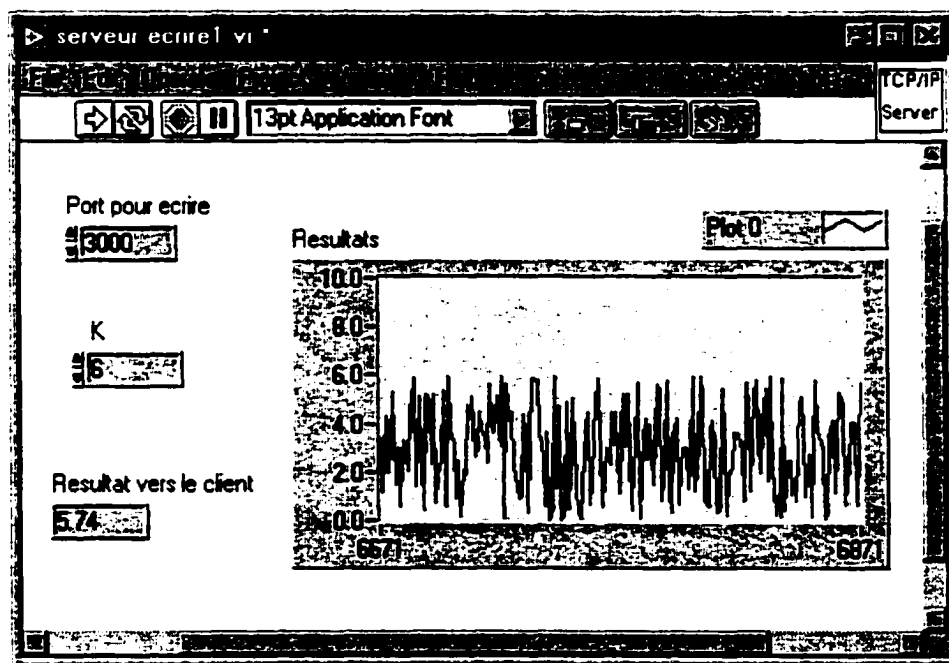


Figure 3. 53 : Serveur qui écrit dans le client.

On a commencé par une simple application client serveur. Dans cette application, c'est le serveur qui fait une tâche quelconque, puis il envoie les résultats vers le client, et ce dernier, ne fait que recevoir et afficher les données reçues. Voir figures 3.53 et 3.54.

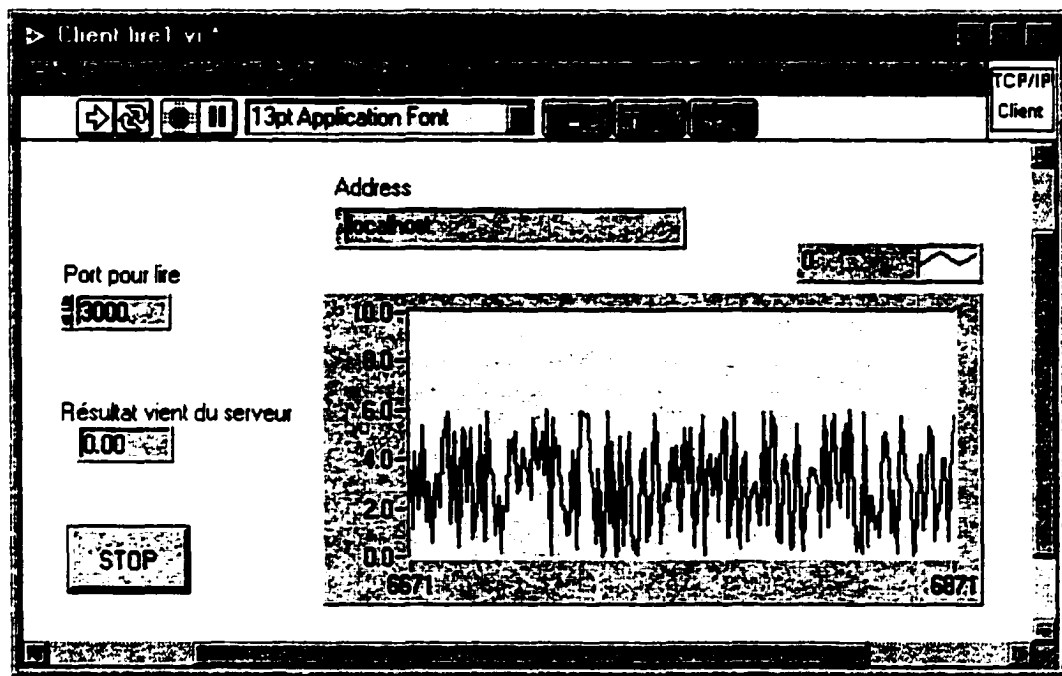


Figure 3. 54 : Client qui reçoit du serveur.

Dans la deuxième application, on a inversé les rôles. Maintenant c'est le client qui envoie une valeur vers le serveur. Ce dernier reçoit la valeur et l'affiche. Comme on peut voir dans les figures 3.55 et 3.56. On peut remarquer dans le serveur le délai pendant la transition d'un niveau à un autre, comme on peut remarquer la différence dans la base de temps entre le serveur et le client, il y avait un délai de quelques secondes pendant la communication. Ce délai varie selon l'état du réseau et on n'a pas pu le contrôler.

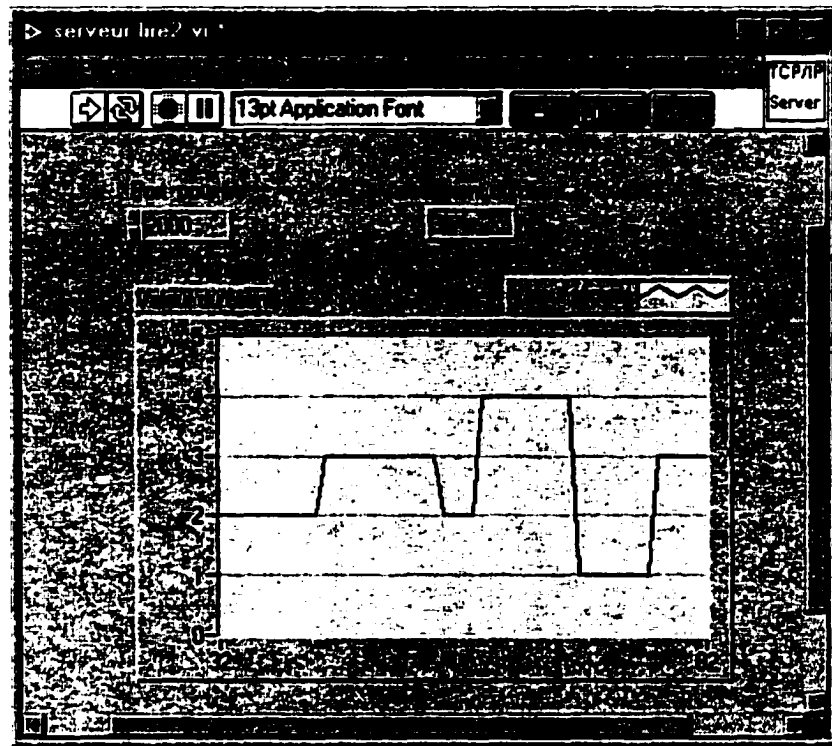


Figure 3. 55 : Serveur reçoit du client.

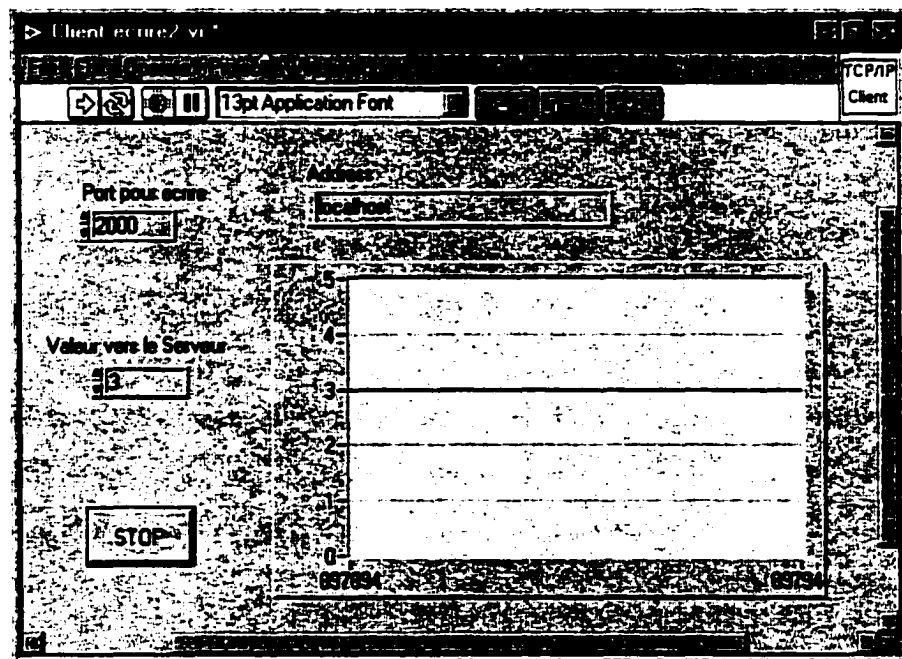


Figure 3. 56 : Client envoie vers le serveur.

Une troisième application est faite, plus intéressante que les autres. Dans celle-ci, c'est le client qui envoie une valeur vers le serveur. Ce dernier fait une tâche quelconque, puis retourne les résultats de nouveau vers le client qui à son tour reçoit ces résultats et les affiche.

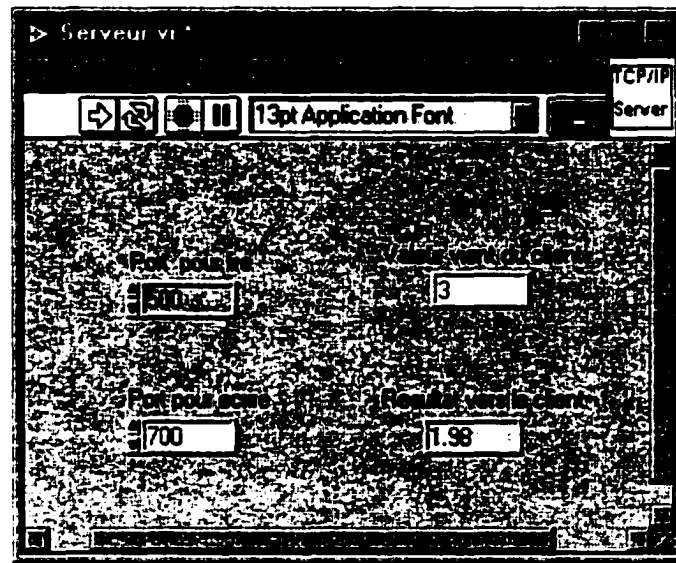


Figure 3. 57 : Serveur reçoit puis il renvoie vers le client.

Pour cette application, on avait besoin de deux connections, figures 3.57 et 3.58, et pour chacune on avait besoin d'un port.

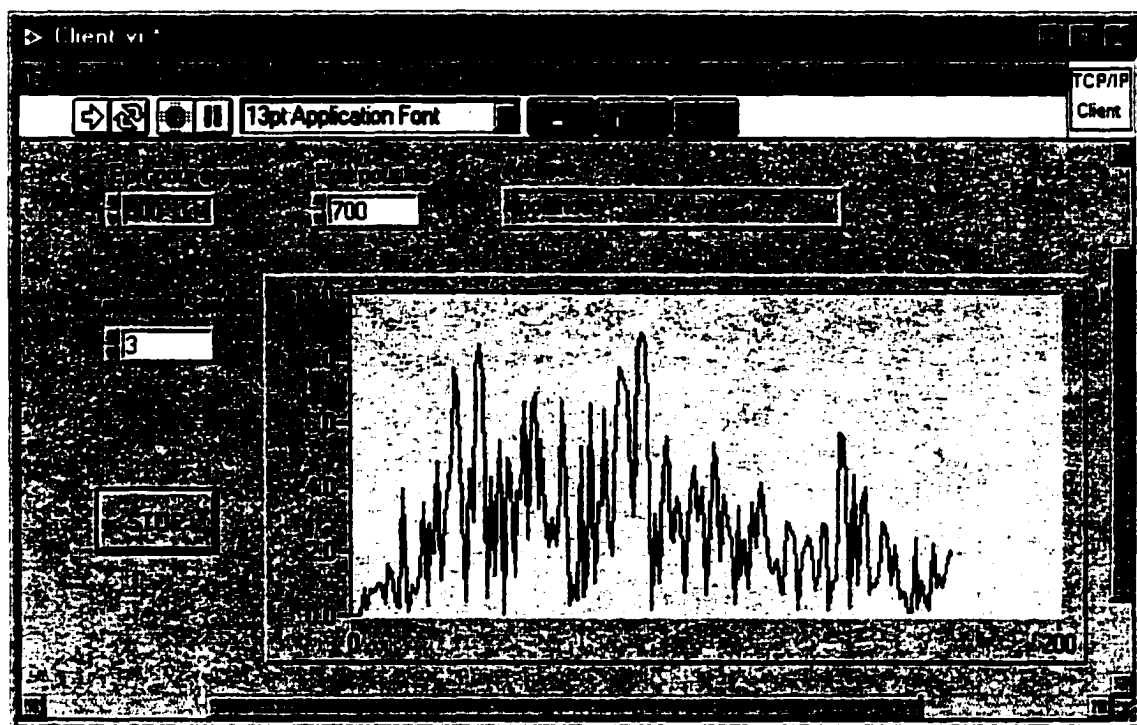


Figure 3. 58 : Client qui envoie puis il reçoit du serveur.

Avec LabVIEW, on peut faire la communication entre plusieurs clients et un serveur, toujours avec le protocole TCP/IP. Il faut spécifier pour chaque client un port de communication. Et si chaque client veut lire et écrire en même temps, dans ce cas, il faut avoir deux connections par client, ça veut dire deux ports. Mais il faut faire en sorte qu'un seul client puisse prendre le contrôle et les autres ne peuvent qu'observer.

Dans cet aspect, on a réalisé une autre application client-serveur, mais cette fois avec deux clients à la fois. Pendant l'exécution les deux clients doivent être présents, parce que le serveur attend un signal de chacun d'eux. Si un client est absent alors on ne peut pas continuer l'exécution. Mais il y a toujours une solution : On peut avoir une copie de son programme dans la machine du serveur ou de l'autre client, alors il suffit de le lancer sur la même machine où le serveur est lancé ou sur une machine voisine,

comme ça le serveur reçoit le signal approprié et on peut continuer l'exécution. Notons qu'avec LabVIEW on peut lancer plusieurs applications sur la même machine.

3.17.4 Contrôle du moteur à distance

Pour cela on a fait aussi une application client-serveur. Le serveur contient l'application où on contrôle le moteur, sauf qu'on a ajouté un programme pour accepter la commande à distance et le client est une autre application qui communique avec le serveur en utilisant le protocole TCP/IP.

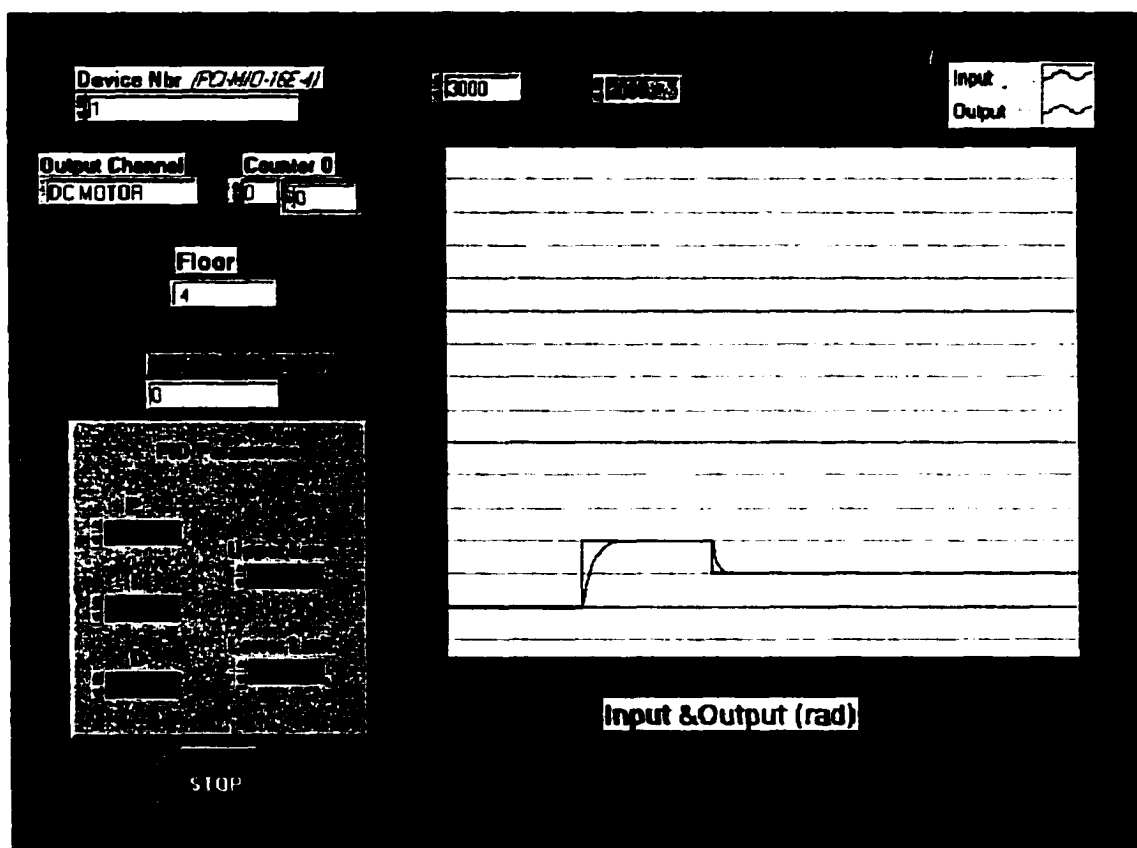


Figure 3. 59 : Application serveur qui contrôle le moteur.

Dans cette application il faut spécifier l'adresse (IP) du serveur. Il faut spécifier aussi le numéro du port à travers lequel on fait la communication. Voir figure 3.59 et 3.60.

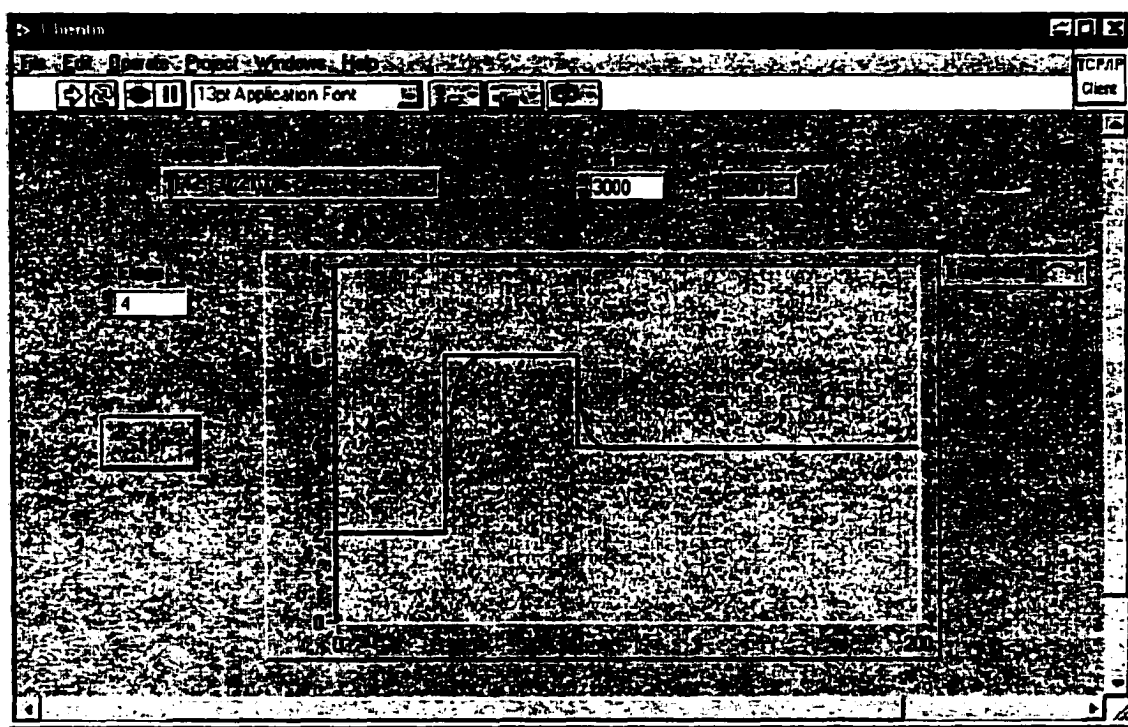


Figure 3. 60 : L'application client qui permet de contrôler le moteur à distance.

Le client envoie par la suite une commande vers le serveur (moteur) qui est en effet la consigne de position (Étage). Puis il reçoit le résultat, qui est la position réelle du moteur. Voir figure 3.61. Dans le serveur, on fait toujours l'acquisition de données à travers la carte d'acquisition, en plus on fait la régulation avec le régulateur PID et on fait la communication avec le client. On peut ajouter les paramètres du PID dans la partie client, mais ça devient un peu compliqué parce que pour chaque paramètre il faut ajouter une connexion et un port, ce qui est très difficile avec l'état actuel du réseau.

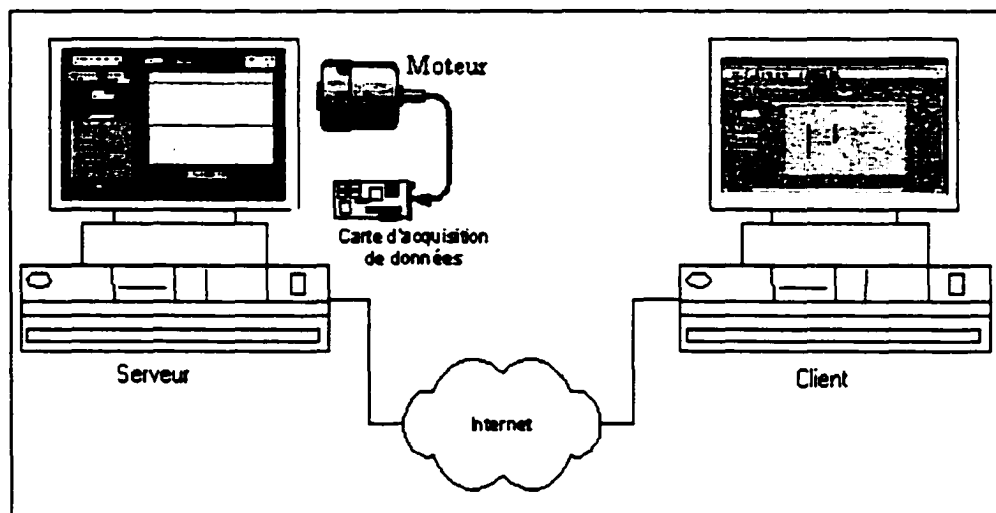


Figure 3. 61 : Application client serveur.

3.18 Conclusion

On a fait des tests entre l'ÉTS et Télé université pour vérifier les applications client- serveurs. On a utilisé un autre logiciel (Remote Administrator) pour visualiser l'écran qui est à l'ÉTS et on a eu des bons résultats, même pour l'application qui manipule le moteur à distance. Le serveur qui manipule le moteur directement était à l'ÉTS et le client était à Télé université, le test s'est bien déroulé; on a pu contrôler le moteur et on a pu changer sa position.

Les résultats étaient acceptables, bien que quelques fois on a rencontré quelques problèmes dus au réseau. Donc il faut penser à optimiser les programmes d'un côté et à améliorer la qualité du réseau d'un autre côté. Il faut que les réseaux utilisés soient dédiés à cet effet pour obtenir de meilleurs résultats et pour pouvoir ajouter d'autres paramètres comme le PID.

Dans notre cas, on contrôle un moteur dont la charge est un ascenseur. Alors on a deux cas complètement différents. Si l'ascenseur monte ce n'est pas la même chose que

s'il descend; il faut donc que les paramètres du PID soient différents dans les deux cas.
Comme ça on peut aussi améliorer le contrôle du moteur.

CHAPITRE 4

COMMANDE À DISTANCE D'UN SIMULATEUR DE PROCÉDÉS

4.1 Introduction

Ce chapitre présente l'accès à distance via le réseau d'Internet d'un simulateur de procédés permettant un contrôle de débit, de pression ou de niveau. Une description de ce simulateur est donnée dans la section 4.2. Dans la section 4.3, l'architecture du laboratoire virtuel nous permettant l'accès au banc d'essai est expliquée. Dans la section 4.4, nous expliquons la communication entre LabVIEW et l'équipement physique par les ports RS232 et RS485. Les résultats de cette application sont donnés dans la section 4.5. Finalement une conclusion suit dans la section 4.6.

4.2 Description du simulateur

Avant d'expliquer le fonctionnement du simulateur, nous allons examiner quelques éléments essentiels nécessaires pour le bon fonctionnement du simulateur et en particulier les transmetteurs.

Tout transmetteur se comporte fondamentalement comme un instrument de mesure dont la particularité est d'élaborer des signaux standard. Pour ce faire, il a besoin d'une source d'alimentation : la figure 4.1 schématise un transmetteur avec :

Le signal d'entrée transmis par le capteur en contact avec le procédé;

Le signal de sortie standard;

La source d'alimentation.

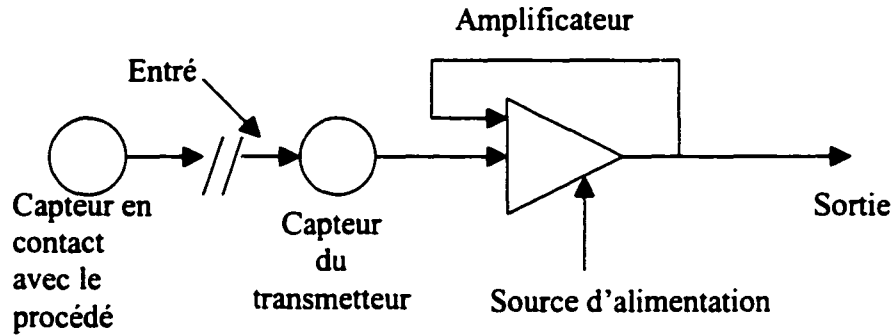


Figure 4. 1 : Principe de fonctionnement du transmetteur.

Le capteur ou le transducteur propre du transmetteur reçoit le signal recueilli par le capteur en contact avec le procédé et le transforme en un signal adapté au dispositif qui élabore le signal de sortie. Le dispositif est appelé amplificateur et il est l'élément principal du transmetteur.

Une étape d'étalonnage est requise avant l'utilisation du transmetteur. Cet étalonnage est applicable aux transmetteurs de pression relative, absolue ou différentielle; dans ce cas, la pression d'étalonnage est appliquée du côté de la haute pression, le côté de la basse pression étant ouvert à l'atmosphère (voir figure 4.2).

La procédure à suivre pour calibrer le simulateur est la suivante :

- 1- Effectuer le montage nécessaire approprié tel qu'il est illustré dans la figure 4.3. La pression d'air d'étalonnage est mesurée par un manomètre à déformation élastique.
- 2- Effectuer les connexions électriques : alimentation et ampèremètre.

3- Mettre le circuit électrique sous tension.

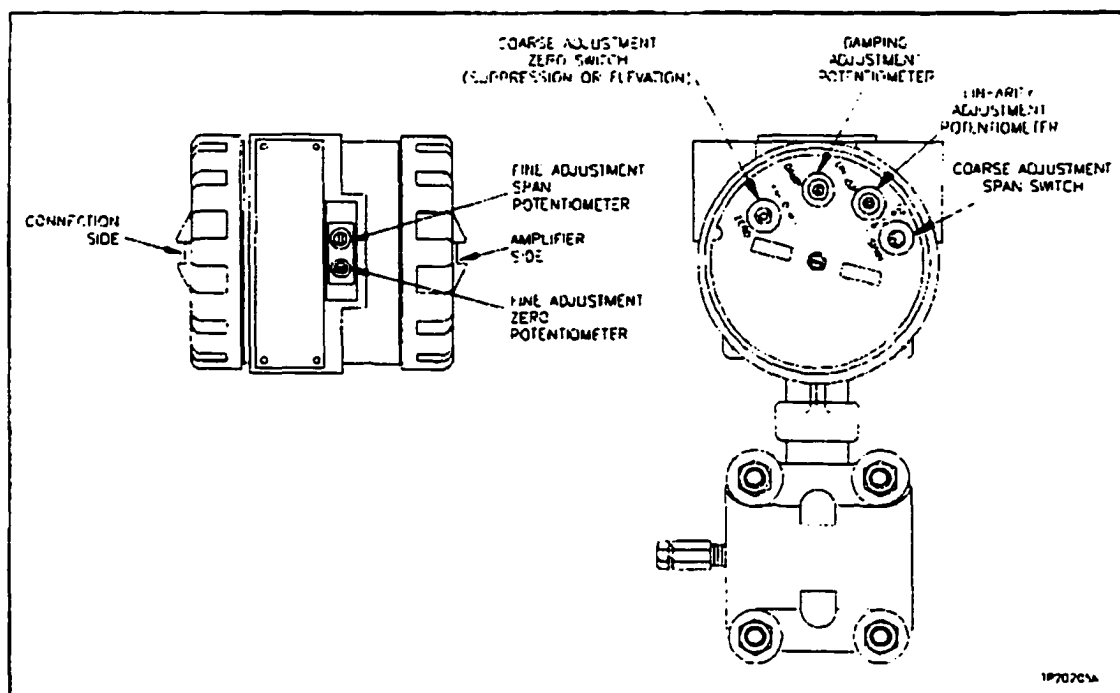


Figure 4. 2 : Les ajustements interne et externe du transmetteur

- 4- Appliquer une pression égale à celle de la portée minimale de l'entendue considérée.
- 5- Si le signal de sortie n'est pas égal à 4.0 mA plus ou moins la tolérance acceptable, agir sur le dispositif de réglage du zéro pour l'obtenir.
- 6- Augmenter la pression d'étalonnage à la valeur de la portée maximale et agir sur le dispositif de réglage de l'étendue jusqu'à l'obtention de la valeur 20.0 mA plus ou moins la tolérance acceptable. (Selon le fabricant, le dispositif de réglage de l'entendue est identifié sur l'appareil par R (Range) ou S (Span)).

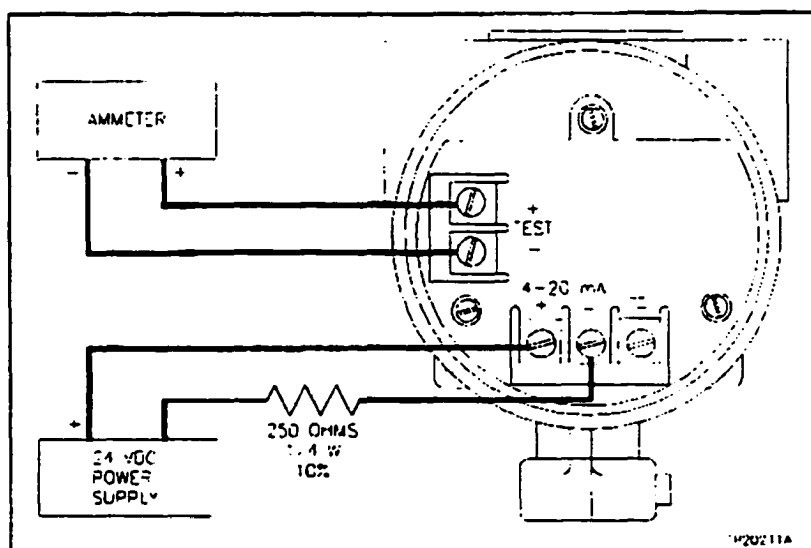


Figure 4. 3 : Montage de calibrage des transmetteurs de pression.

7- Vérifier le zéro de nouveau en appliquant la pression nécessaire. Faire l'ajustement. Dans certaines conceptions, les deux dispositifs de réglage du minimum et du maximum sont inter- reliés; Le réglage de la portée maximale a une influence non négligeable sur celui du zéro qu'on doit donc réajuster.

8- Appliquer une pression correspondante à 50% de l'entendue d'échelle ; Si le signal de sortie n'est pas égal à 50% de son échelle (12.0 mA), agir sur l'éventuel dispositif de linéarité pour l'amener à cette valeur de 12.0mA.

Ce même transmetteur est à la base des boucles de niveau, de débit et de pression. Il nous donne un signal de 4 à 20 mA proportionnel à la variable à contrôler. Si nous souhaitons connaître le niveau, la pression ou le débit, il suffit de mesurer le courant et de le convertir dans les unités appropriées. La figure 4.4 nous montre une vue globale du simulateur.

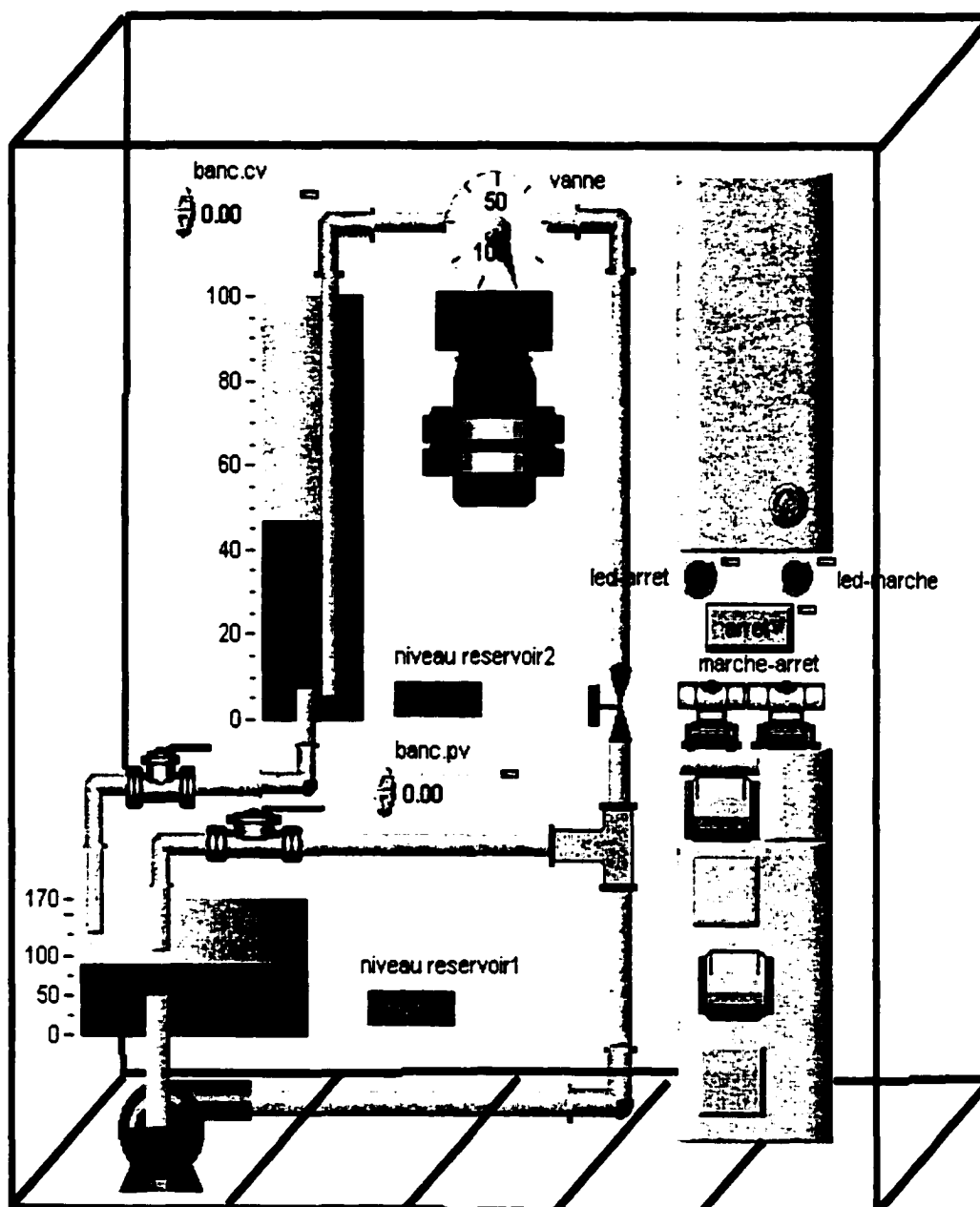


Figure 4. 4 : Simulateur de procédés.

4.3 Architecture du laboratoire virtuel

Avant de procéder à la commande du simulateur à distance, une configuration pour commande locale a été mise en place pour interagir en temps réel avec le simulateur. La figure 4.5 montre les éléments essentiels de cette configuration.

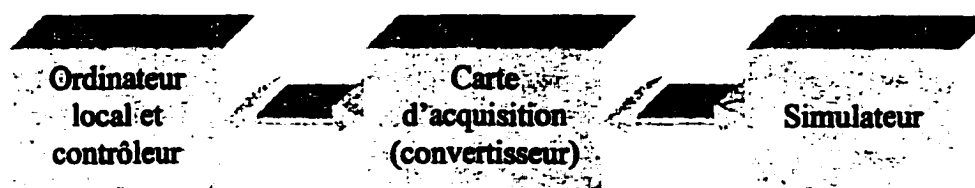


Figure 4. 5 : Configuration locale

Comme nous l'avons expliqué dans le chapitre 1, la carte d'acquisition utilisée est de type PCI-MOI-16E-4 de National Instrument. Notons que ce simulateur est muni de deux automates de Fisher-Bailey nous permettant de communiquer par les ports séries RS232 et RS485. En se servant de LabVIEW, nous avons testé les 3 modes mentionnés ci-haut. Nous nous limiterons pour le moment à l'utilisation de la carte d'acquisition et dans la prochaine section, nous présenterons l'utilisation des ports RS232 et RS485.

Pour transmettre les signaux sur le réseau d'Internet et accéder aux différents paramètres de ce simulateur, deux possibilités s'offraient. La première est d'utiliser les utilitaires de LabVIEW et la seconde est d'utiliser Microsoft NETMEETING. La première possibilité est tout à fait fonctionnelle en multipoint en ce qui concerne les signaux autres que vidéos, par contre la deuxième nous donne accès à tous les signaux mais fonctionne point à point. Même si les deux approches ont été utilisées et des résultats satisfaisants ont été obtenus, nous limiterons notre présentation à l'approche utilisant NETMEETING. Les signaux vidéos ont été transmis en utilisant une caméra

PROSHARE. Une vue du simulateur telle que transmise par la caméra est illustrée à la figure 4.6

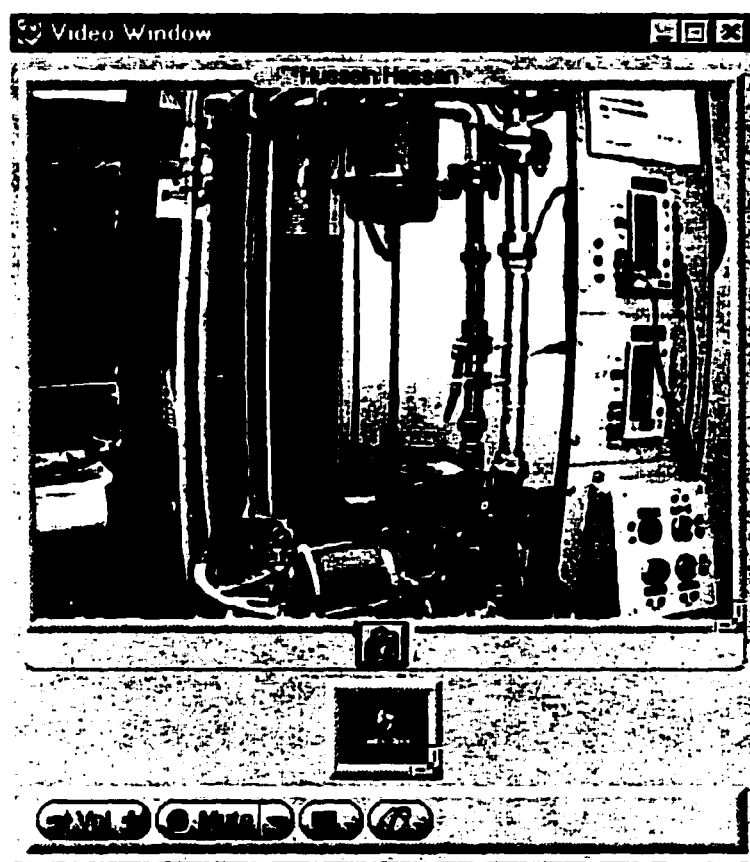


Figure 4. 6 : Simulateur de procédés vu par la caméra PROSHARE.

Une fois l'architecture est en place, nous avons procédé à la conception de l'interface LabVIEW nous permettant de manipuler le simulateur via le réseau Internet. Deux types de manipulations sont possibles. Le premier est en boucle ouverte nécessaire pour l'identification des paramètres du simulateur. Ces paramètres sont en fait les paramètres du modèle dynamique du simulateur. Le deuxième type d'interaction est en boucle fermée pour l'asservissement des variables de contrôle. Pour le besoin d'illustration, nous avons choisi de présenter le contrôle du niveau. L'interface LabVIEW utilisée pour interagir localement et à distance avec le banc d'essai est

illustrée à la figure 4.7. Nous pouvons constater du côté gauche de la figure une représentation schématique du niveau à contrôler. En fait, cette représentation illustre le fonctionnement global du simulateur. En temps réel, localement ou à distance, le niveau du réservoir est affiché donnant une idée exacte à l'utilisateur du niveau atteint. Cette figure peut être vue simultanément avec l'image transmise par la caméra. Du côté droit de la figure 4.7, nous pouvons voir tous les paramètres à manipuler ou à modifier. Il y a en particulier les paramètres du contrôleur : les gains proportionnel, intégral et dérivé. Il y a aussi les paramètres de consigne et des limites physiques à respecter. Ces derniers paramètres nous assurent une certaine sécurité de débordement pour éviter l'arrêt fréquent du système. Cette fonctionnalité est très importante pour une manipulation à distance. Le bas du côté droit est réservé aux signaux temporels du niveau, de la commande et du signal de référence.

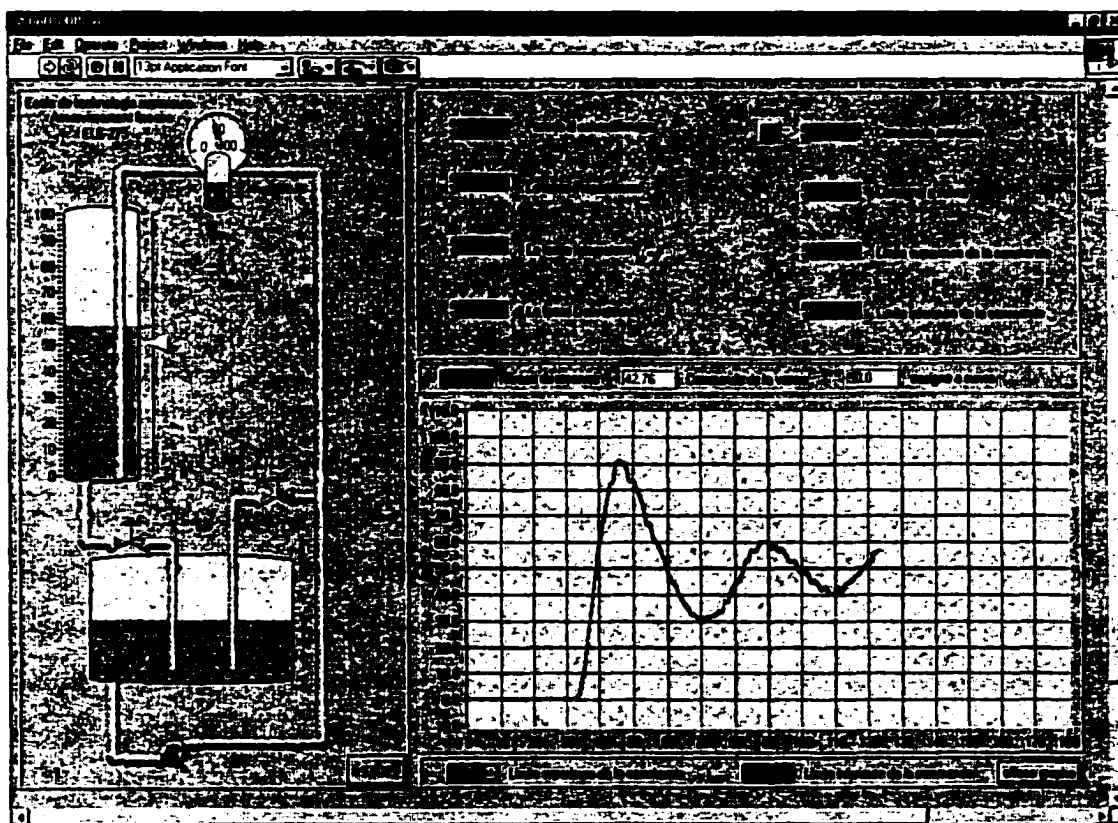


Figure 4. 7 : Interface LabVIEW du simulateur.

4.4 Communication avec les ports RS232 et RS485

Comme nous l'avons mentionné dans la section précédente, le simulateur est équipé d'automates Fisher-Bailey. Une communication série RS232 est disponible en plus de la communication avec un port RS485. La première est seulement point à point tandis que la seconde est multipoint. Ainsi avec le RS485, nous avons la possibilité de faire communiquer plusieurs contrôleurs sur le même bus de données comme l'illustre la figure 4.8.

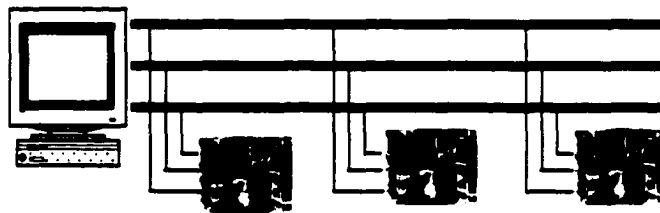


Figure 4. 8 : Illustration de la communication RS485.

Rappelons qu'avec ces deux modes, la caméra PROSHARE est aussi utilisée pour transporter le signal vidéo sur le réseau. Notons à ce stade que les pilotes (drivers) pour les automates Fisher-Bailey ne sont pas fournis par LabVIEW. Nous étions obligés d'écrire ces pilotes en langage C car LabVIEW nous permet d'appeler un programme écrit dans un langage de programmation conventionnel, comme le langage C par exemple. On peut à partir du diagramme LabVIEW utiliser le CIN (Code Interface Node) pour exécuter des tâches que les langages de programmation conventionnels peuvent réaliser plus rapidement que LabVIEW.

La façon de le faire est de définir d'abord les paramètres du CIN en particulier toutes les entrées et les sorties. Nous créons par la suite un programme C qui contient les mêmes paramètres déjà définis pour le CIN. Nous compilons le programme pour créer un fichier "obj" qui sera par la suite modifié pour générer un fichier "lsb" avec un programme fourni avec LabVIEW qui s'appelle "lvsbutil.exe". En chargeant le fichier

"lsb" dans le CIN, on sera prêt à exécuter le programme LabVIEW reconnaissant les adresses physiques en entrées et en sorties du simulateur. Une explication plus détaillée sur la façon de créer un CIN pour LabVIEW est donnée en Annexe G.

4.5 Application

Dans cette section, nous présentons le simulateur en fonctionnement et les résultats obtenus. La figure 4.9 nous montre en même temps l'interface LabVIEW et le simulateur de procédés transmis par la caméra. Dans cette image, c'est la carte d'acquisition qui est en fonction. Nous pourrions voir que l'utilisateur peut visualiser le système ainsi que l'interface. Tous les paramètres du simulateur sont accessibles à distance. Même deux partenaires peuvent faire l'expérience tout en étant chacun d'un côté. La seule règle à respecter c'est au niveau de contrôle simultanée, i.e. une variable ne peut être contrôlée simultanément par plus qu'une personne. Pour partager l'accès à la machine, il faut transférer le contrôle aux autres partenaires à tour de rôle.

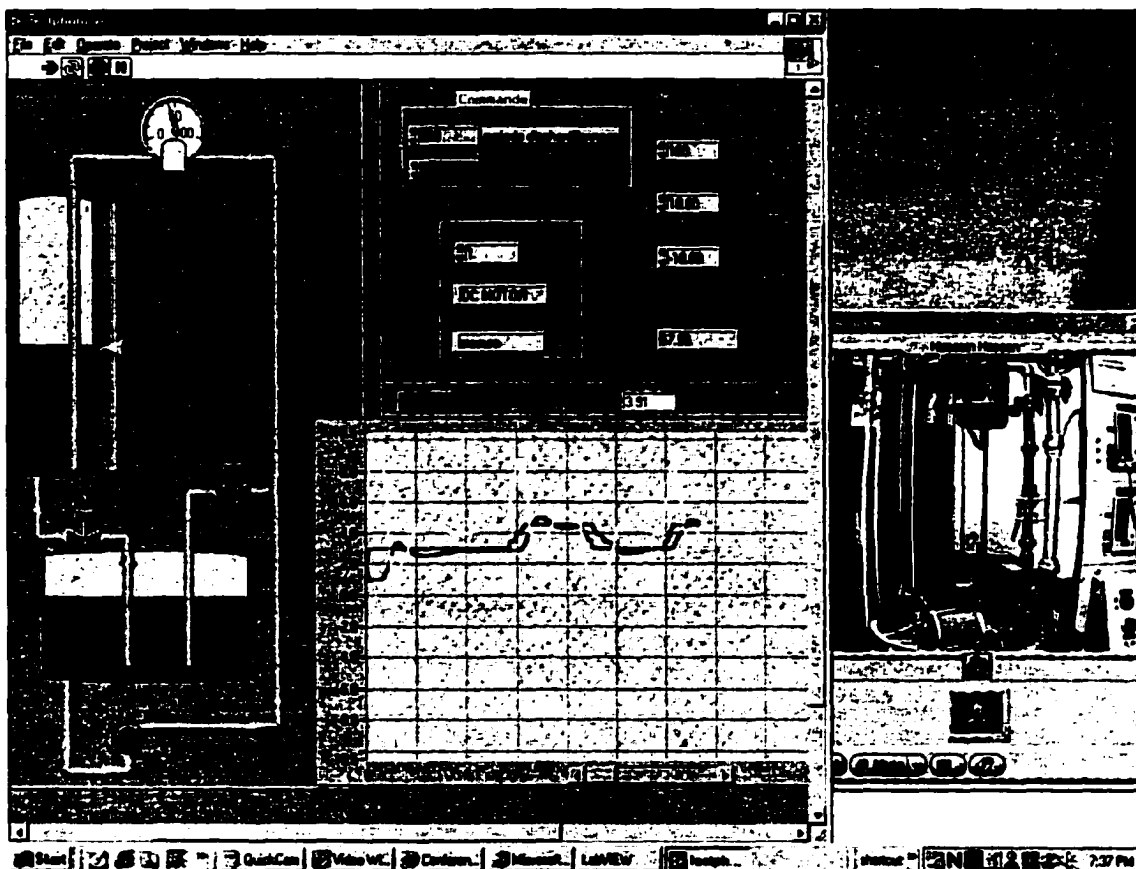


Figure 4. 9 : Vue d'ensemble du simulateur.

Notons aussi que la première étape est celle reliée à l'identification de la dynamique du procédé. Cette étape est faite en boucle ouverte et peut être réalisée à distance sans aucun problème. Pour ce faire, nous utilisons généralement le mode manuel. Ce mode est identifié sur le simulateur comme étant le mode manuel. Il est basé sur l'analyse de la courbe de réponse à un échelon unité. C'est la courbe de variation du niveau e VP suite à une variation brusque et entretenue de la grandeur régulante, en boucle ouverte comme le montre la figure 4.10.

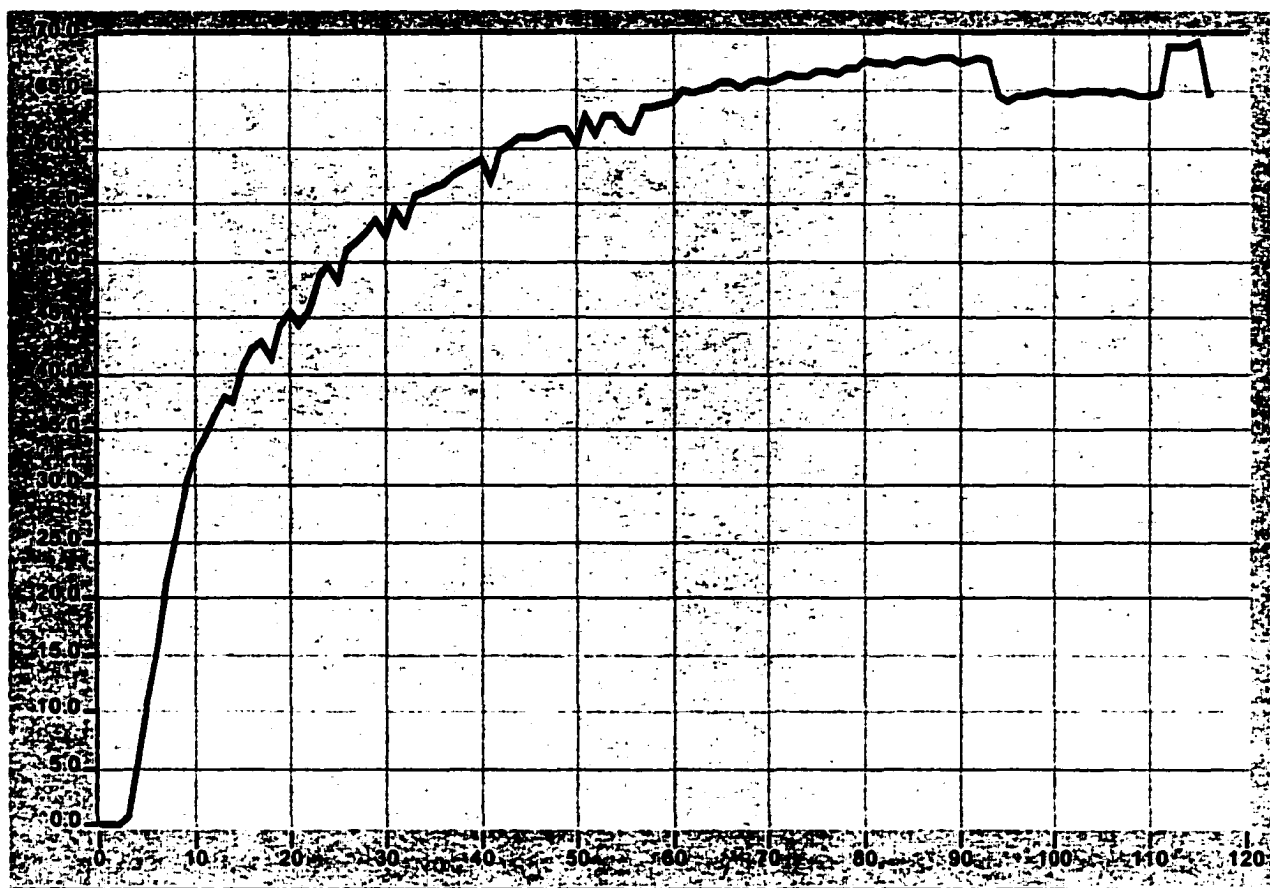


Figure 4. 10 : Réponse à l'échelon

La réponse à l'échelon nous indique que le système est de premier ordre. Le gain statique est le rapport de la valeur en régime permanent sur la grandeur de l'échelon appliquée. La constante de temps est le temps correspondant à 63% de la sortie en régime permanent.

Il est facile de déterminer dans ces conditions le gain statique et la constante de temps de la dynamique du système.

4.6 Conclusion

Ce chapitre nous a montré l'accès à distance à un simulateur de procédés regroupant le contrôle de niveau, de débit et de pression. Nous avons réalisé cette application en intégrant LabVIEW avec une carte d'acquisition et en communiquant avec les automates via RS232 ou RS485. Les pilotes des automates ont été écrits en langage C. La transmission sur le réseau Internet a été faite avec NETMEETING. Les résultats obtenus sont très satisfaisants. Malheureusement la transmission actuelle ne peut être faite que point à point. Notons finalement que les problèmes de délai ne sont pas nuisibles étant donné que le contrôle est implanté localement. L'utilisateur interagit avec une image transmise sur le réseau Internet. Dans le cas de problèmes de délai, l'accès peut être coupé momentanément avec le système mais le contrôle reprendra correctement par la suite.

CONCLUSION

L'apprentissage du logiciel de développement d'application LabVIEW s'est fait tout d'abord de manière générale avec la découverte des énormes potentialités de cet outil informatique et ensuite, il a été orienté vers le système du moteur d'ascenseur et le simulateur de procédés à réaliser. L'assimilation a été plus ou moins bonne.

En modélisant le moteur à courant continu en un système linéaire, on se rapproche des données réelles même si des différences peuvent surgir à cause des paramètres de simplification et des approximations faites.

Bien que l'étude ait été faite pour un moteur à courant continu en particulier, le cheminement est tout aussi valable lorsqu'on change les valeurs des paramètres dynamiques.

La sortie qui caractérise le mouvement d'ascenseur est sensiblement égale à l'entrée qui est assimilée dans notre cas à l'étage où on désire se rendre, grâce à l'introduction d'un contrôleur dans l'application.

La réalisation en temps réel a renforcé mes connaissances acquises aussi bien au niveau du logiciel que pour la commande du système.

L'utilisation du logiciel LabVIEW et l'application des principes de l'asservissement pour commander le moteur et le simulateur de procédés m'ont permis

d'approfondir mes connaissances dans les informations des commandes industrielles et de bien comprendre et savoir comment gérer les processus industriels.

Dans un deuxième temps j'ai pu découvrir les avantages des outils industriels. L'utilisation de l'Internet comme une source documentaire et un aide technique m'est apparue très efficace.

Pour améliorer le contrôle en temps réel, on doit penser à préparer un environnement plus fiable à celle qu'on avait, un système d'exploitation autre que Windows, il faut qu'il soit spécifié pour le contrôle en temps réel; que les matériels utilisés soient dédiés à cet effet, un micro ordinateur très puissant et où il roule seulement les programmes nécessaires pour faire marcher l'application, une carte d'acquisition plus perfectionnée. On est arrivé à un temps d'échantillonnage de 20 ms, ce qui est considéré très bon, mais on vise contrôler le système à travers le réseau, dans ce cas il faut penser à améliorer le système au complet.

On a atteint les objectifs du projet, on a pu contrôler, à distance et en temps réel, un moteur à courant continu et un simulateur de procédé. Pour la communication entre machine-PC, on a testé trois modes différents tels que : une carte d'acquisition et les deux ports séries RS232 et RS485. Pour la communication à distance, on a utilisé le protocole TCP/IP et Microsoft NETMEETING et on a ajouté une camera pour transmettre des images vidéo à travers le réseau. Les résultats étaient acceptables. L'utilisation du logiciel LabVIEW était très avantageuse surtout avec ses outils intégrés comme le protocole TCP/IP et l'acquisition de données. On peut penser maintenant à écrire un scénario d'un laboratoire qui utilise le moteur ou le simulateur et voir comment on peut le réaliser à distance.

ANNEXE A

LABVIEW

LABVIEW

Qu'est-ce que LabVIEW ?

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) est un environnement de développement basé sur le langage de programmation graphique G. LabVIEW est entièrement intégré pour assurer une communication avec le matériel *GP/IB*, *VXI*, *PXI*, *RS-232*, *RS-485* ainsi que les cartes d'acquisition de données. LabVIEW dispose aussi de bibliothèques intégrées conçues pour les standards logiciels comme la gestion de données TCP/IP et ActiveX.

Grâce à LabVIEW, on peut créer des programmes compilés 32 bits qui vous offrent les vitesses d'exécution nécessaires pour des solutions d'acquisition de données, de test et mesure personnalisées. On peut également créer des exécutables autonomes étant donné que LabVIEW est un véritable compilateur 32 bits.

On peut se servir de LabVIEW sans grande expérience préalable en programmation. En effet, LabVIEW utilise une terminologie, des icônes et des concepts familiers aux techniciens, scientifiques et ingénieurs. En outre, il s'inspire de symboles graphiques plutôt que du langage textuel pour décrire des opérations de programmation.

Les programmes de LabVIEW, destinés aux applications physiques, sont divers et variés. Ils peuvent être très simples ou au contraire très puissants, comme l'illustre le graphique suivant.

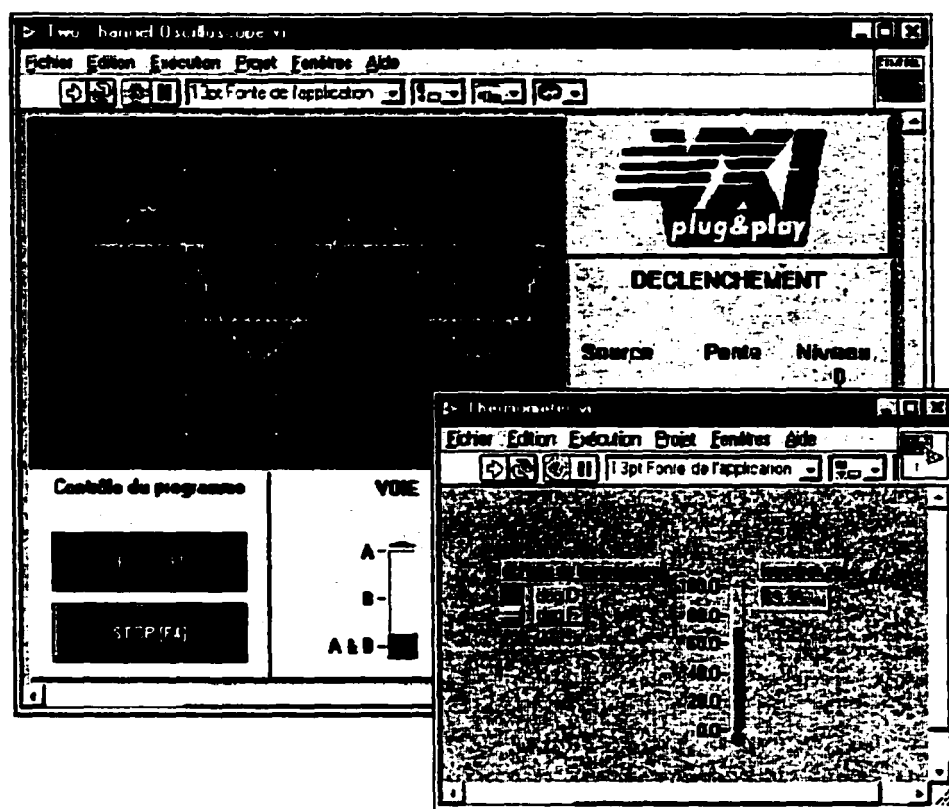


Figure A. 1 : La programmation avec LabVIEW.

LabVIEW contient des bibliothèques étendues spécialisées dans la collecte, l'analyse, la présentation et le stockage de données. Il intègre en outre une panoplie complète d'outils de développement de programmes conventionnels, de sorte qu'on peut définir des points d'arrêt, animer l'exécution du programme pour visualiser son fonctionnement et exécuter pas à pas notre programme. Le développement et la mise au point du programme s'en trouvent ainsi facilités.

LabVIEW offre également un grand nombre de mécanismes spécialisés dans la connexion au code externe ou à des logiciels via des DLL, des bibliothèques partagées

ou ActiveX, pour n'en citer que quelques-uns. En outre, on aura à notre disposition une grande variété de boîtes à outils complémentaires pour satisfaire tous nos besoins.

Quel est l'intérêt de LabVIEW ?

LabVIEW nous donne la possibilité de construire nos propres solutions pour les systèmes scientifiques et techniques. Il nous offre la souplesse et les performances des puissants langages de programmation sans les difficultés et la complexité qui les caractérisent.

LabVIEW offre désormais à des milliers d'utilisateurs un moyen rapide de programmer des systèmes d'instrumentation et d'acquisition de données.

En utilisant LabVIEW pour prototyper, concevoir, tester et mettre en œuvre nos systèmes d'instrumentation, on peut réduire le temps imparti au développement du système et ainsi augmenter la productivité d'un facteur 4 à 10.

LabVIEW nous fait bénéficier d'une importante base d'utilisateurs en ligne, d'années d'informations sur les produits ainsi que de puissants outils complémentaires. Enfin, le réseau des utilisateurs et le support technique de National Instruments garantissent le succès du développement de nos solutions.

Comment fonctionne LabVIEW ?

Tous les programmes de LabVIEW, communément appelés *instruments virtuels* (VIs), se composent d'une *face-avant* et d'un *diagramme*. Les *palettes* de LabVIEW contiennent un certain nombre d'options qui nous servent à créer et à modifier nos VIs.

Cette section présente la face-avant, le diagramme et les palettes de LabVIEW. Elle fait aussi un tour d'horizon de l'exécution des programmes par *flux de données* propre à LabVIEW.

La face-avant

La face-avant est l'interface utilisateur graphique de notre VI LabVIEW. Cette interface reçoit les données entrées par l'utilisateur et affiche celles fournies, en

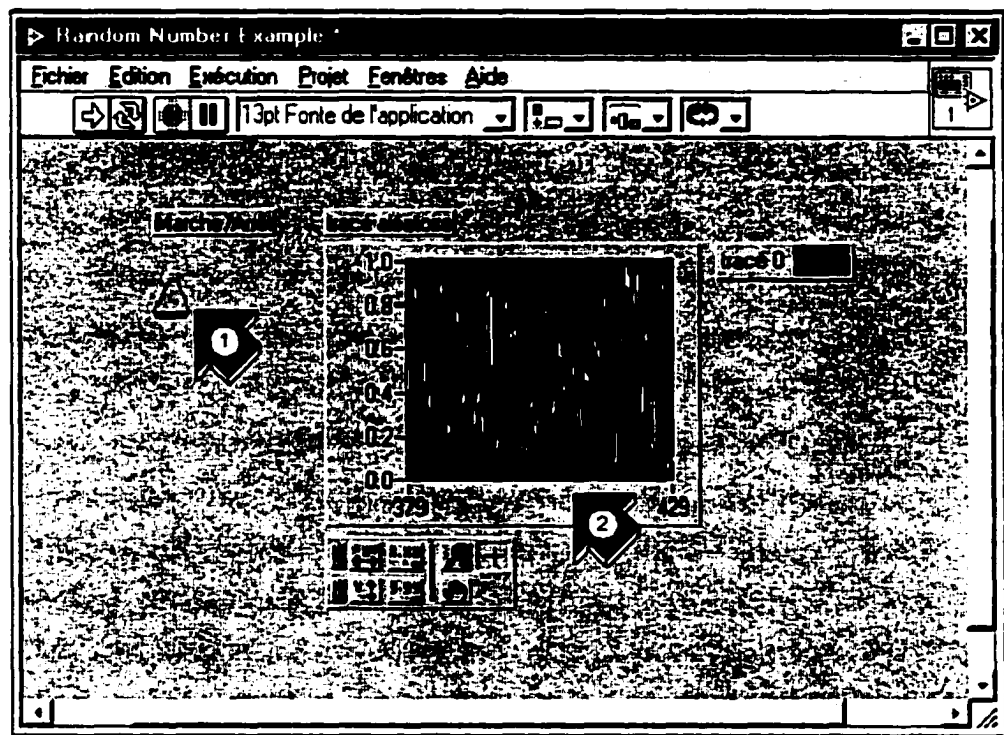


Figure A. 2 : Interface avec LabVIEW.

sortie, par le programme. Cette face-avant peut contenir des boutons rotatifs, des boutons-poussoirs, des graphes, et d'autres sortes de commandes ou d'indicateurs.

1. Une *commande* (entrée).
2. Un *indicateur* (sortie).

Le diagramme

Le diagramme contient le code source graphique de notre VI. On y programme notre VI pour contrôler et remplir des fonctions sur les entrées et sorties créées dans la face- avant.

Le diagramme peut contenir des *fonctions* et des *structures* issues des bibliothèques de VIs intégrées à LabVIEW. Il peut aussi contenir des *terminaux* associés à des commandes et à des indicateurs créés dans la face-avant.

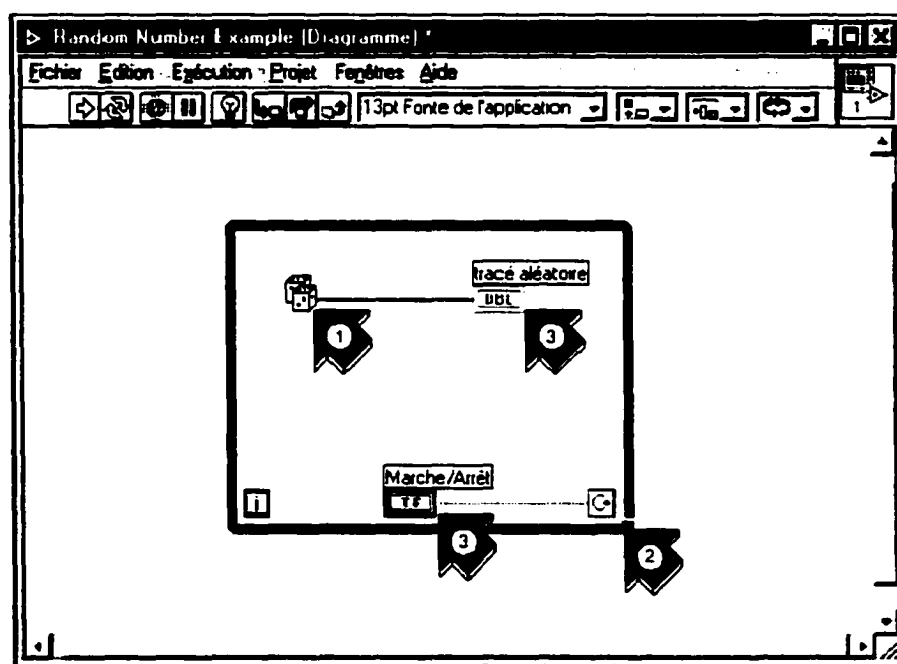


Figure A. 3 : La face diagramme.

Dans cette figure on a :

1. Une fonction.
2. Une structure.
3. Terminaux de la face-avant.

Les palettes

Les palettes de LabVIEW nous offrent les options dont on a besoin pour créer et éditer la face-avant et le diagramme :

1- La palette d'outils

On utilise la palette *d'outils* aussi bien dans la face-avant que dans le diagramme. Elle contient les outils nécessaires pour éditer et mettre au point les objets du diagramme et de la face-avant.



Figure A. 4 : la palette d'outils.

2- La palette de commandes

On utilise la palette de commandes dans la face-avant uniquement. Elle contient les commandes et les indicateurs de face-avant qui vous servent à créer l'interface utilisateur.

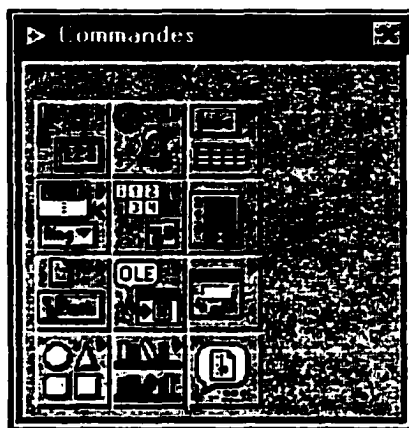


Figure A. 5 : la palette de commandes.

3- La palette de fonctions

On utilise la palette de fonctions dans le diagramme uniquement. Elle contient les objets qui servent à programmer notre VI, comme par exemple des opérations arithmétiques, d'E/S d'instrument, d'E/S sur fichiers et d'acquisition de données.

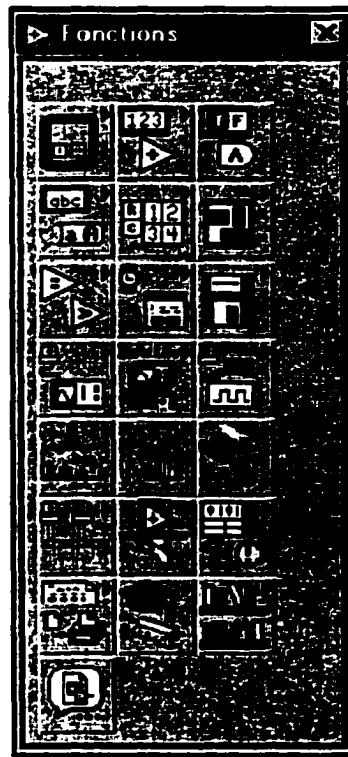


Figure A. 6 : La palette de fonctions.

Le flux de données

Les VIs de LabVIEW suivent le concept de l'exécution de programme par flux de données. Le diagramme se compose de *nœuds*, tels que des VIs, des structures ou des terminaux de la face-avant. Ces nœuds sont reliés entre eux par des *fils*, qui définissent le flux de données à travers le programme.

L'exécution d'un nœud n'a lieu que lorsque toutes ses entrées sont disponibles. Dès qu'un nœud a fini de s'exécuter, il transfère toutes ses sorties au prochain nœud dans le chemin de flux de données.

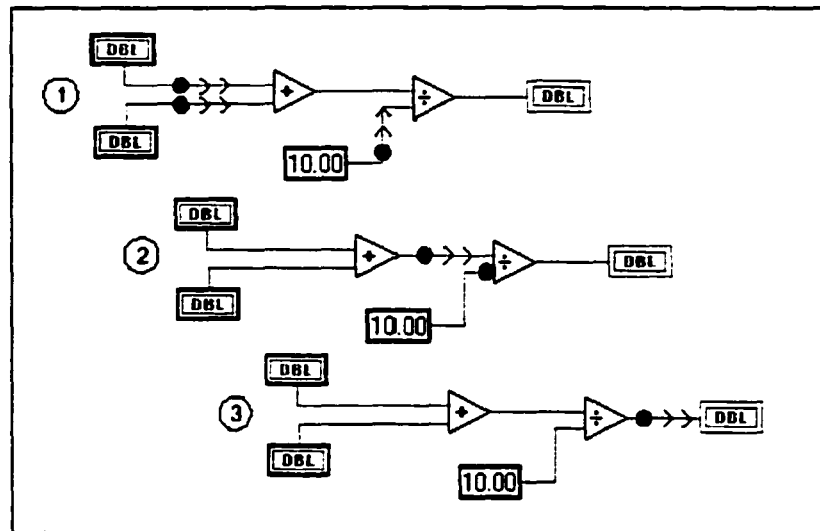


Figure A. 7 : Les flux de données dans le diagramme LabVIEW.

ANNEXE B

L'encodeur incrémental

L'encodeur incrémental

Principe de fonctionnement

Les encodeurs incrémentaux sont destinés à des applications de positionnement et de contrôle de déplacement d'un mobile par comptage et décomptage des impulsions qu'ils délivrent. Voir figure B.1.

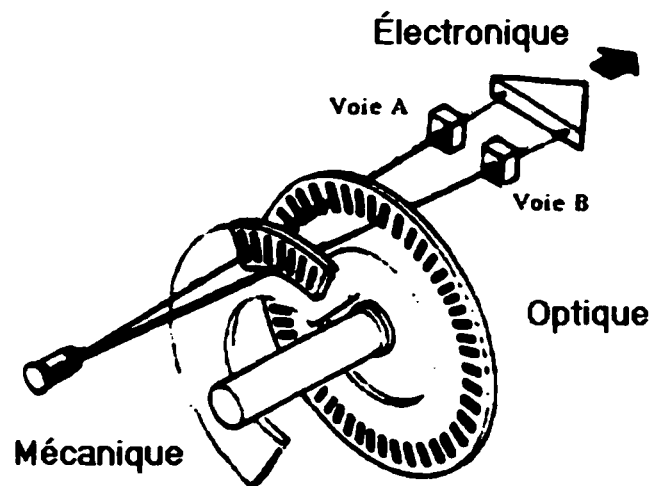


Figure B. 1: Encodeur incrémental.

Le disque d'un encodeur incrémental comporte deux types de pistes :

La piste extérieure: (voie A ou voie A et B) est divisée en « n » intervalles d'angles égaux alternativement opaques et transparents, « n » s'appelant **la résolution** ou nombre de périodes. C'est en effet le nombre d'impulsions qui seront délivrées par l'encodeur pour un tour complet de son disque.

Deux photodiodes décalées sont installées derrière la piste extérieure, ils délivrent des signaux carrés A et B en quadrature.

La piste intérieure : (voie 2) comporte une seule fenêtre transparente. Celle-ci ne délivre donc qu'un seul signal par tour. Ce signal Z appelé «top zéro» dure 90° électriques et est synchrone des signaux A et B. Ce «top zéro» détermine une position de référence et permet la réinitialisation à chaque tour.

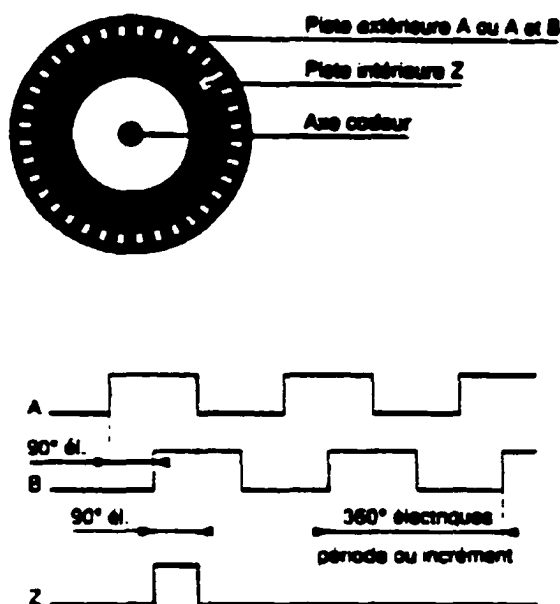


Figure B. 2 : Les impulsions dans un encodeur incrémental.

Détermination du sens de rotation

Le déphasage de 90° électrique des signaux A et B, figure B. 3, permet de déterminer le sens de rotation :

Dans un sens pendant le front montant du signal A, le signal B est à 0.

Dans l'autre sens pendant le front montant du signal A, le signal B est à 1. Figure B. 4.

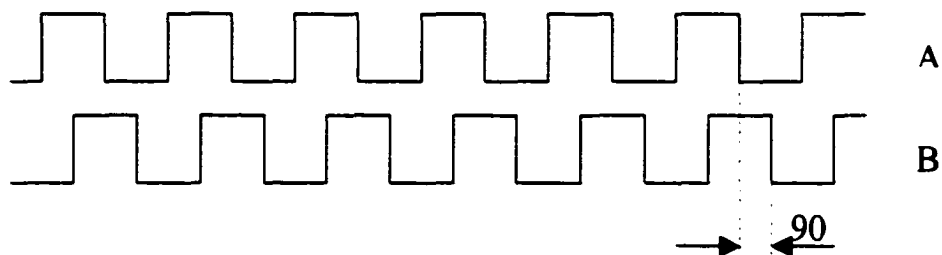


Figure B. 3 : Déphasage entre les deux signaux A et B.

L'utilisation d'un encodeur incrémental nécessite une mise à zéro du compteur à sa mise sous tension.

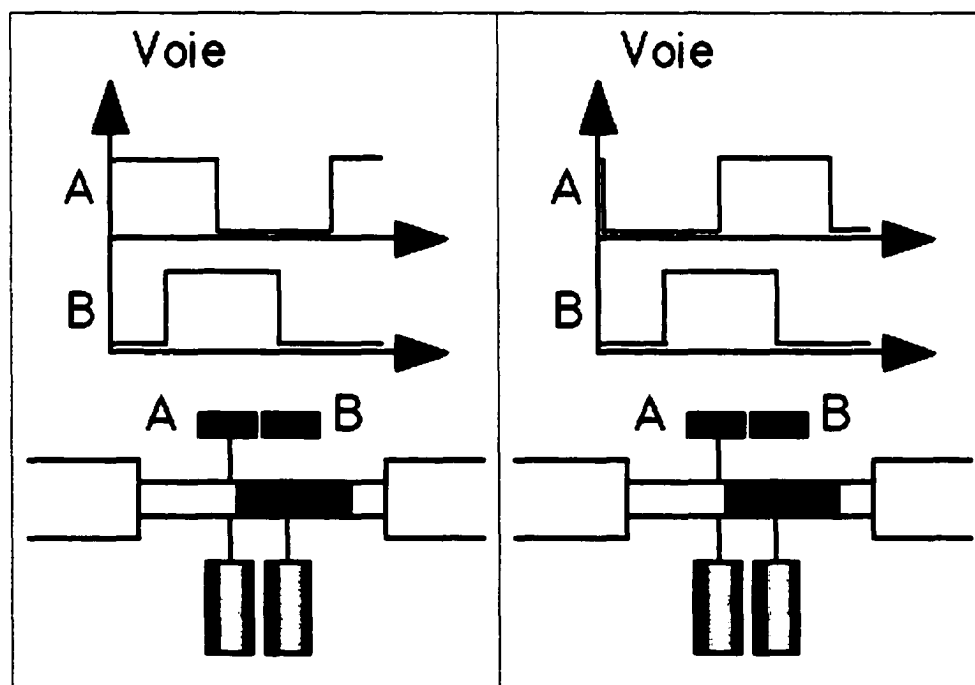


Figure B. 4 : Sens de rotation.

Exploitation des voies A et B

Fréquemment un traitement électronique (intégré à un encodeur) permet de délivrer les signaux complémentaires A, B, et Z. Nous disposons donc à la sortie de l'encodeur incrémental de six signaux A, A, B, B, Z, Z qui autorisent trois niveaux de précision d'exploitation :

Utilisation des fronts montants de la voie A seule : Exploitation simple correspondant à la résolution de l'encodeur.

Utilisation des fronts montants et descendants de la voie A seule : La précision d'exploitation est doublée.

Utilisation des fronts montants et descendants des voies A et B : La précision d'exploitation est quadruplée.

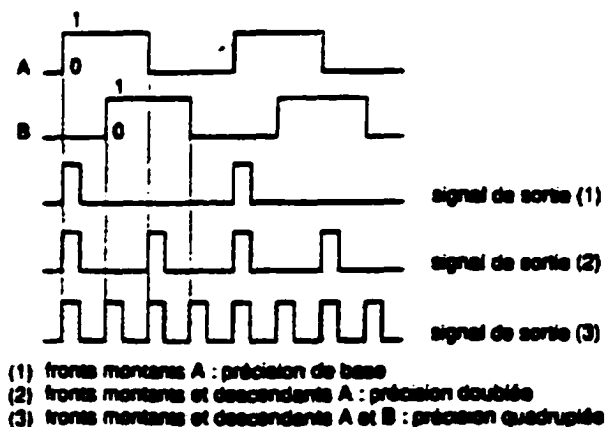


Figure B. 5 : Précision.

Élimination des parasites

L'apparition des parasites en ligne peut perturber le comptage des impulsions délivrées par l'encodeur incrémental. En effet, ces parasites seront parfois comptabilisés de la même façon que les impulsions délivrées par l'encodeur.

Si le système de traitement est conçu pour pouvoir exploiter les différents signaux A , \bar{A} , B , \bar{B} , Z , \bar{Z} délivrés par l'encodeur incrémental il sera possible de différencier les impulsions encodeur des impulsions parasites évitant ainsi la prise en compte de ces dernières.

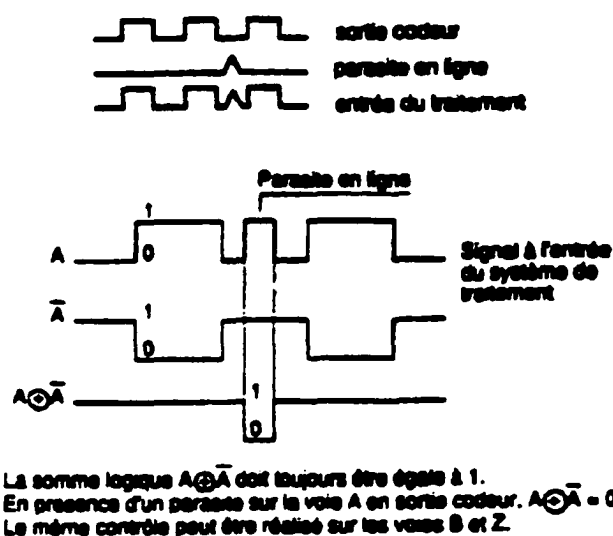


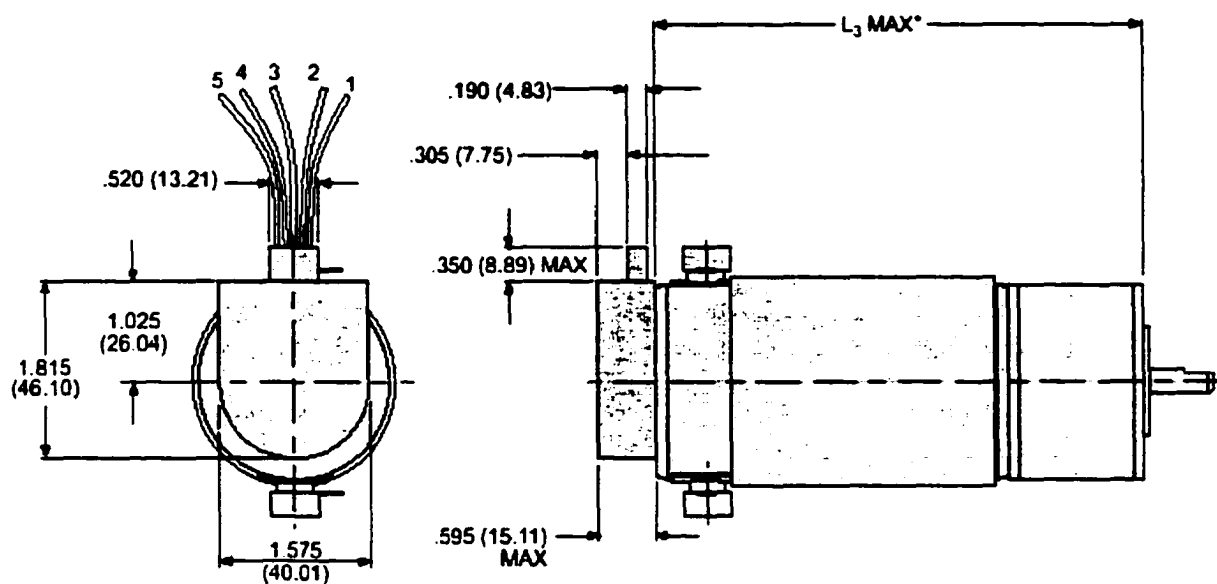
Figure B. 6 : Parasites.

Nous avons choisi un encodeur incrémental pour notre application car il a une conception simple (son disque ne comporte que deux pistes) donc plus fiable et moins onéreux qu'un encodeur absolu.

ANNEXE C

Moteur et encodeur

Moteur et encodeur



Encoder Connection Chart

Pin No.	Color	Connection
1	Black	Ground
2	Green	Index/NC
3	Yellow	Channel A
4	Red	Vcc
5	Blue	Channel B

PITTMAN®

Figure C. 1 : Moteur et encodeur.

ANNEXE D

Schéma interne de la carte d'acquisition

Schéma interne de la carte d'acquisition

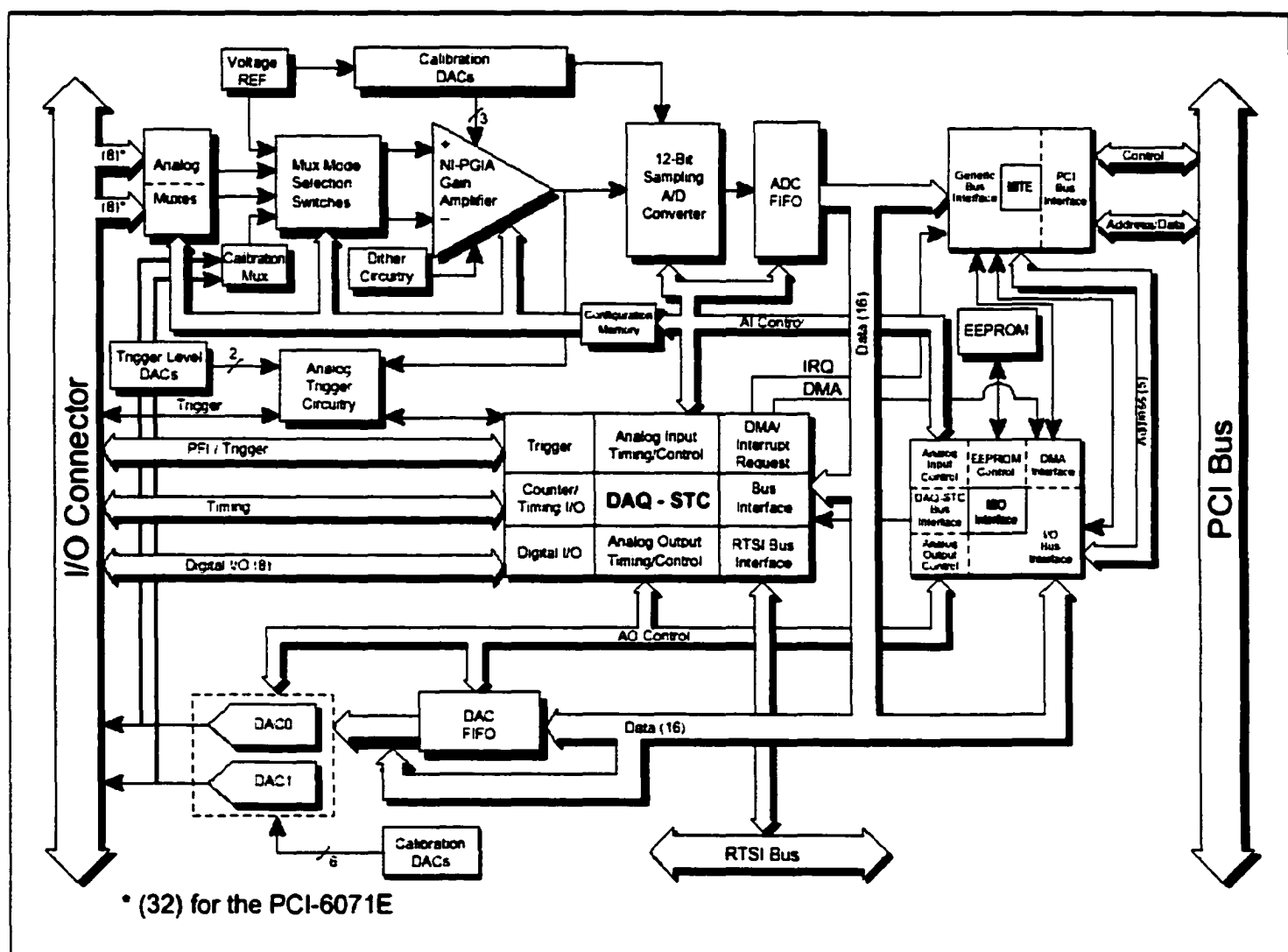


Figure D. 1 : Schéma interne de la carte d'acquisition PCI-MOI-16 E-4.

ANNEXE E

Les E/S connexions pour la carte PCI-MOI-16 E-4

Les E/S connexions pour la carte PCI-MOI-16 E-4

ACH8	34	68	ACH0
ACH1	33	67	AIGND
AIGND	32	66	ACH9
ACH10	31	65	ACH2
ACH3	30	64	AIGND
AIGND	29	63	ACH11
ACH4	28	62	AISENSE
AIGND	27	61	ACH12
ACH13	26	60	ACH5
ACH6	25	59	AIGND
AIGND	24	58	ACH14
ACH15	23	57	ACH7
DAC0OUT1	22	56	AIGND
DAC1OUT1	21	55	AOGND
EXTREF2	20	54	AOGND
DIO4	19	53	DGND
DGND	18	52	DIO0
DIO1	17	51	DIO5
DIO6	16	50	DGND
DGND	15	49	DIO2
+5 V	14	48	DIO7
DGND	13	47	DIO3
DGND	12	46	SCANCLK
PF10/TRIG1	11	45	EXTSTROBE*
PF11/TRIG2	10	44	DGND
DGND	9	43	PF12/CONVERT*
+5 V	8	42	PF13/GPCTR1_SOURCE
DGND	7	41	PF14/GPCTR1_GATE
PF15/UPDATE*	6	40	GPCTR1_OUT
PF16/WFTRIG	5	39	DGND
DGND	4	38	PF17/STARTSCAN
PF19/GPCTR0_GATE	3	37	PF18/GPCTR0_SOURCE
GPCTR0_OUT	2	36	DGND
FREQ_OUT	1	35	DGND

Figure E. 1 : Les E/S connexions pour la carte d'acquisition.

ANNEXE F

Les programmes avec Matlab

Les programmes avec Matlab

Contrôle du moteur sans PID avec Simulink :

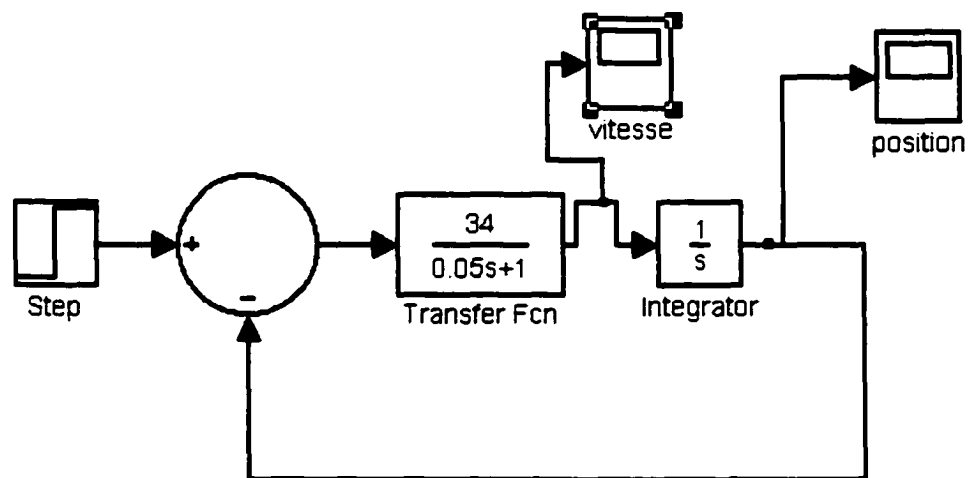


Figure F. 1 : Boucle fermée avec Simulink.

%Position et Vitesse en boucle ouverte :

```
T=0.01; % période d'échantillonnage
K=54;   % gain statique
```

```

Tm=0.04; % constante de temps

a1=1+exp(-T/Tm);
a2=exp(-T/Tm);
a3=K*(T-Tm*(1-a2));
a4=K*(T*a2-Tm*(1-a2));

Kf=floor(1/T);

%passe 1
o(1) = 0;
u(1)=3;
vit(1)=0;

%passe 2
o(2) = a2*o(1)+a3*u(1);
u(2) = 3;
vit(2)=(o(2)-o(1))/T;

%boucle
for k=3:Kf+1,
    u(k)=3;
    o(k) = a1*o(k-1)-a2*o(k-2)+a3*u(k-1)-a4*u(k-2);
    vit(k)=(o(k)-o(k-1))/T;
end;

t=(0:Kf)*T;

ysup=u(1)*ones(t)*(1+0.05);
yinf=u(1)*ones(t)*(1-0.05);
%subplot(211)
plot(t,o);
title('Position o'), xlabel('temps(sec)'), ylabel('rad')
%subplot(212)
grid
figure
plot(t,vit);
title('vitesse'), xlabel('temps(sec)'), ylabel('rad/sec')
grid
----
```


Test_avec PD en boucle fermée :

```

clear
T=0.01; % période d'échantillonnage
K=85; % gain statique
Tm=0.1; % constante de temps
Kp=1; % gain propotionnel
Kd=0; % gain derive

a1=1+exp(-T/Tm);
a2=exp(-T/Tm);
a3=K*(T-Tm*(1-a2));
a4=K*(T*a2-Tm*(1-a2));

x=(a4+(a4*(a4-a3))^0.5)/a3;

%Kp=(a2-1)*x^2/a4;
%Kd=-T*a2*x^2/a4;

A1=Kp+Kd/T;
A0=-Kd/T;

r=3;
Kf=floor(.6/T);

%passe 1
o(1) = 0;
e(1) = r - o(1);
u(1) = A1*e(1);
vit(1)=0;
%passe 2

o(2) = a2*o(1)+a3*u(1);
e(2) = r - o(2);
u(2) = A1*e(2)+A0*e(1);
vit(2)=(o(2)-o(1))/T;
%boucle
for k=3:Kf+1,
o(k) = a1*o(k-1)-a2*o(k-2)+a3*u(k-1)-a4*u(k-2);
e(k) = r - o(k);
u(k) = A1*e(k)+A0*e(k-1);
vit(k)=(o(k)-o(k-1))/T;
end;

```

```

t=(0:Kf)*T;

Tr=-4.744*T/log(x);
ysup=r*ones(t)*(1+0.05);
yinf=r*ones(t)*(1-0.05);
%subplot(211)
plot(t,o);
title('Position o'), xlabel('temps(sec)'), ylabel('rad')
%subplot(212)
grid
figure
plot(t,vit);
title('Vitesse'), xlabel('temps(sec)'), ylabel('rad/sec')
grid
----
```

ANNEXE G

Code Interface Node (CIN)

Code Interface Node (CIN)

Appeler un code externe avec LabVIEW

Lorsqu'on ouvre un VI quelconque, LabVIEW compile le code graphique source G et il le transforme en un langage machine (langage d'assembleur) de sorte que lorsqu'on lance le programme avec le bouton RUN, on exécute un programme déjà compilé. Le fichier.exe a été écrit en C, mais les VIs n'impliquent jamais une couche de C, on écrit le code en G (Graphique), et ce code sera compilé directement dans un langage machine, c'est pour cette raison que LabVIEW est tellement rapide.

Quoique LabVIEW exécute son code rapidement, quelques utilisateurs peuvent vouloir appeler un certain code externe. On peut citer trois raisons pour lesquelles les utilisateurs appellent le code externe :

Ils ont un code existant qu'ils veulent l'utiliser dans un VI.

Ils veulent communiquer avec un hardware que LabVIEW ne supporte pas directement.

Ils veulent faire une tâche qu'on ne peut pas réaliser directement à partir du diagramme de LabVIEW.

Avec LabVIEW, il existe plusieurs méthodes qui peuvent être employées pour appeler un code externe écrit avec un autre langage de programmation, parmi ces méthodes :

- Code Interface Node (CIN).
- Call library Function.
- Active X Client Functions.
- System Exec.vi.

Dans notre cas, on s'intéresse seulement à l'utilisation du Code Interface Node (CIN).

Code Interface Node (CIN)

On peut utiliser le Code Interface Node pour appeler un code source écrit dans un langage de programmation basé texte conventionnel. On utilise le CIN pour réaliser des tâches qu'on ne peut pas les faire directement à partir du diagramme ou lorsque les langages de programmation conventionnels peuvent les réaliser plus rapidement que LabVIEW.

Le CIN est une méthode commune d'appel du code C à partir du LabVIEW, on peut faire entrer ou sortir des données complexes en code C. Dans certains cas on obtient de meilleures performances en utilisant des CIN, les structures de données étant transmises au même type que leur stockage dans LabVIEW.

Toutefois, pour obtenir ce niveau de performance, on doit apprendre à créer un CIN. Pour cela, on doit être un bon développeur du C, étant donné que les CIN sont étroitement liés au LabVIEW, on ne peut pas utiliser n'importe quel compilateur.

Les étapes principales pour créer un CIN avec LabVIEW sont les suivantes :

- 1- Définir les paramètres d'entrée et de sortie du CIN.
- 2- Créer le code source (fichier.c).

- 3- Créer le fichier ressource (fichier.lsb).
- 4- Charger le fichier ressource dans le CIN.

1- Définir les paramètres d'entrée et de sortie du CIN

On ouvre le VI avec lequel on travaille ou on ouvre un nouveau VI et on sauvegarde sous un nom désiré. Ce même nom sera le nom par défaut du fichier.c et du fichier ressource.lsb.

Sur la face avant du LabVIEW (*Front panel*), on place les commandes et les indicateurs qu'on veut utiliser comme entrées et sorties. On doit leur donner des noms appropriés et significatifs car ces mêmes noms seront utilisés comme noms des variables dans le code C.

Sur la face arrière (*Block diagram*) on peut voir une représentation pour chaque paramètre placé. On doit ajouter un CIN, que l'on le trouve dans Fonctions Palette → Advanced → CIN. Par défaut dans le CIN on trouve deux paramètres, une entrée et une sortie, on peut ajouter le nombre désiré du paramètre selon les besoins, ceci peut être fait en changeant la dimension du CIN avec l'outil de positionnement (flèche indicatrice) ou en cliquant sur le cadre du CIN avec le bouton droit de la souris. On choisit ensuite « ajouter paramètre » (*Add parameter*). Par la suite il faut lier les variables d'entrées au coté gauche du CIN et les variables de sorties au côté droit.

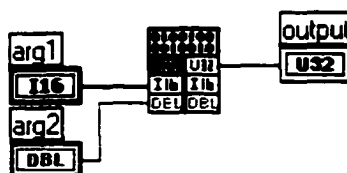


Figure G. 1 : CIN avec ses paramètres.

Il faut assurer que les entrées et les sorties sont du type de données désirées car les variables générées dans le code C vont avoir les mêmes types que les entrées et les sorties. Si dans un cadre de paramètres on veut avoir des résultats seulement (sans entrée) alors il faut changer ce cadre pour sortie seulement (Output Only).

2- Créer le code source (fichier.c).

Pour créer le fichier.c, on ouvre la face arrière (*Block diagram*), on clique avec le bouton droit de la souris sur la partie supérieure (haut) du CIN et on choisit " créer le fichier c " Ceci créera un fichier qui porte par défaut le nom de notre VI. On peut garder le même nom comme on peut le changer puis on le sauvegarde dans un répertoire qui sera utilisé plus tard pour le projet qui génère à son tour les autres fichiers. Le fichier.c généré contient la déclaration de l'entête suivant avant n'importe quel autre code :

```
#include " extcode.h "
```

C'est la librairie principale pour le CIN dans LabVIEW. Si le CIN fait des appels au système alors on doit ajouter l'entête suivant :

```
#include " hosttype.h "
```

Le fichier.c contient aussi le prototype puis la définition de la fonction CIN MgErr CINRun (...). Les paramètres sont les mêmes que ceux déclarés dans le diagramme du VI. Il suffit d'écrire le reste du code à l'endroit spécifié.

3- Créer le fichier ressource (fichier.lsb).

Les étapes spécifiques à suivre pour générer le fichier.lsb dépendent de la plateforme et du compilateur qu'on utilise.

Dans notre cas, on a utilisé Microsoft Visuel C++ IDE (Integrated Developer Environment) pour réaliser ce travail. On décrira en détail toutes les étapes suivies pour créer le fichier.lsb.

a) Créer un projet de DLL :

Créer un nouveau projet en faisant : File → New...

Indiquer le type du projet pour qu'il soit : " Win32 Dynamic-Link Library ".
Voir figure G.2.

Nommer le projet avec le même nom du VI, sans aucun espace. Pour localisation, choisir le répertoire où on a sauvegardé le fichier.c.

Cliquer sur OK.

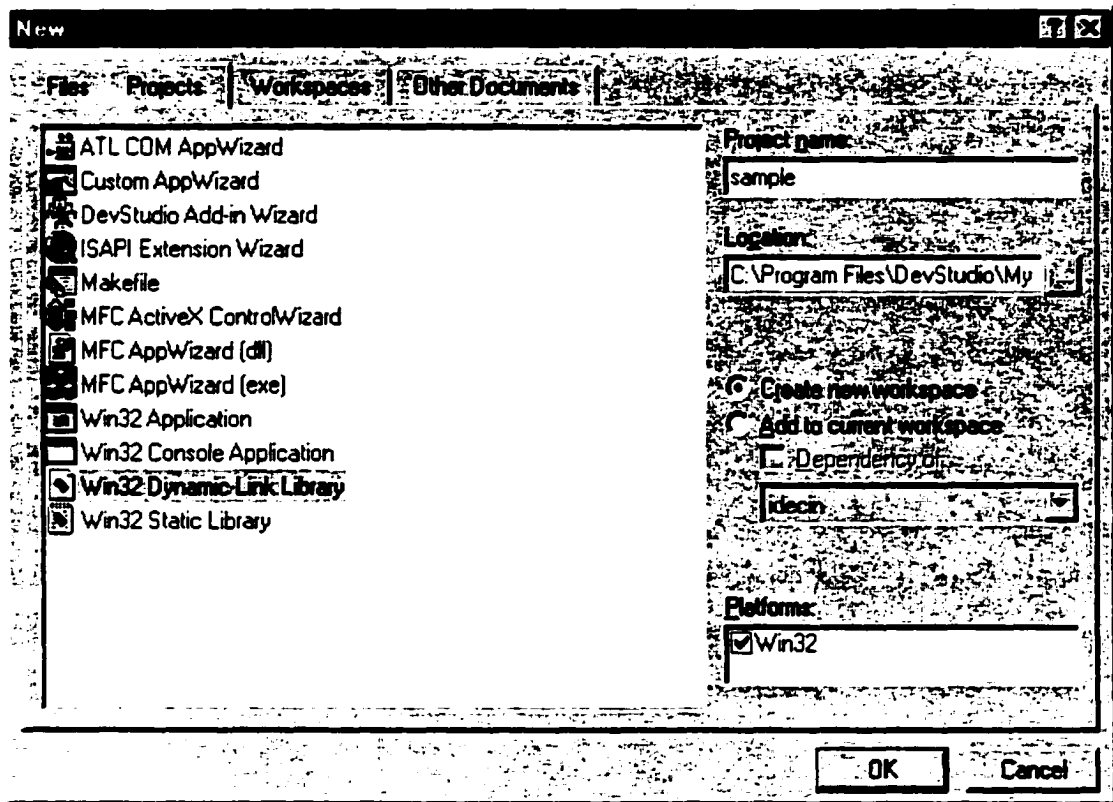


Figure G. 2 : Création d'un nouveau projet.

b) Ajouter les objets et les librairies du CIN au projet :

Choisir Project → Add to Project → Files...

Alors aller au sous-répertoire *Win32*, figure G.3, dans le répertoire *cintools* qui se trouve en général dans :

c : \program files\national Instruments\LabVIEW et choisir les fichiers suivants :

- LabVIEW.lib.
- lvsb.lib.
- lvsbmain.def.
- cin.obj.

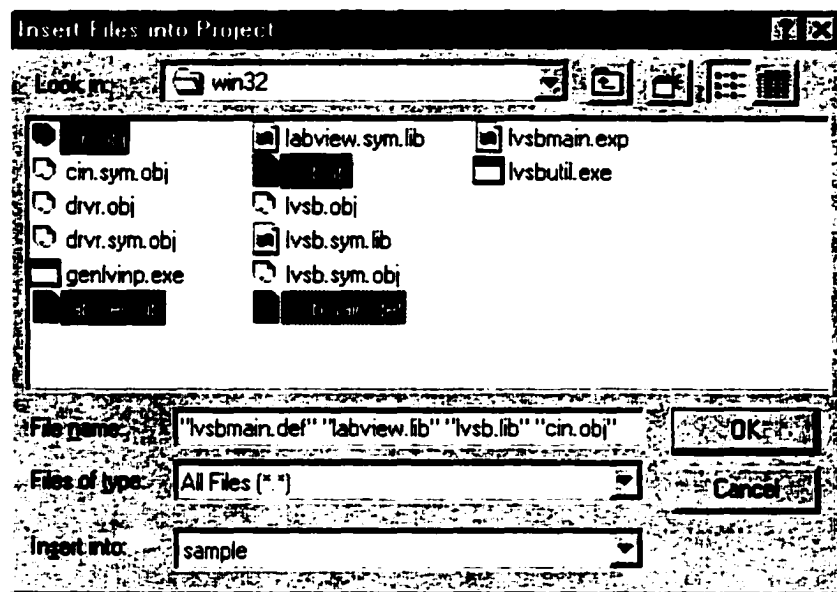


Figure G. 3 : Les fichiers qu'on aura besoin dans le projet.

On peut sélectionner plusieurs fichiers à la fois en maintenant le bouton « Ctrl ». Cliquez alors OK.

c) Ajouter votre fichier.c au projet.

Choisir Project >> Add to Project >> Files...

Aller au répertoire où on trouve le fichier.c que vous avez écrit et le sélectionner.

Cliquez alors OK.

d) Éditer des Configurations de Projet.

Aller au : Project → Settings

Placer le " Settings For " à " All Configurations ".

Toutes les étapes restantes sont dans ce panneau.

Ne cliquez pas le bouton OK, on le fera plus tard.

Dans le tabulateur de " C/C++ "

Placer la zone " Category " au " Preprocessor ", Voir Figure G.4.

Ajouter le chemin d'accès vers le répertoire " cintools " dans la zone " Additional include directories ". Les chemins d'accès doivent être en *MS-DOS Format* (Voir la définition de *MS-DOS Format* plus loin).

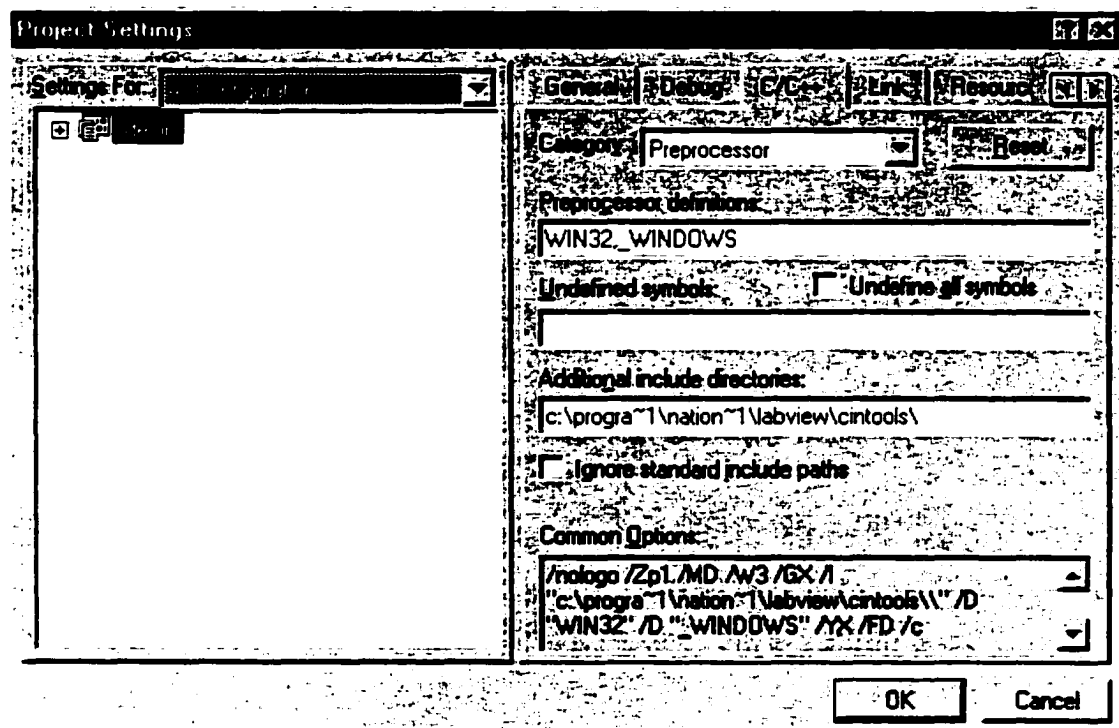


Figure G. 4 : Configuration pour le projet.

Placer la zone " category " au " Code Generation ".

Changer la zone " Use run-time library " en "Multithreaded DLL".

Changer la zone " Struct member alignment " en "1 Byte".

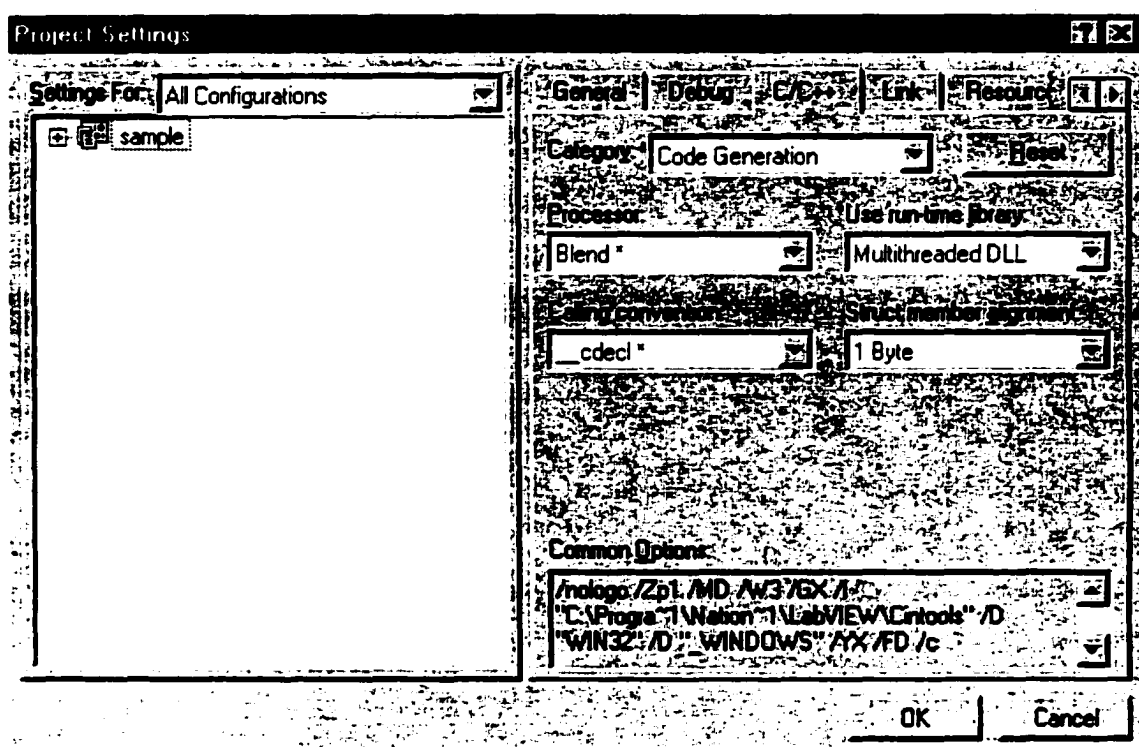


Figure G. 5 : Configuration pour Code Generation.

Dans le tabulateur " Custom Build " (faire défiler vers le droit pour trouver ce tabulateur.) :

Compléter les zones suivantes:

Description : Compléter une description.

Build command(s) : < Le chemin d'accès vers Cintools toujours en *MS-DOS*
Format ... >\win32\lvsbutil \$(Targetname) - d "\$(WkspDir)\\$(OutDir)"

Output file(s) : *\$(OutDir)\\$(TargetName).lsb*. Voir figure G.6.

Cliquer sur le bouton OK.

Note : *Targetname* c'est le nom désiré du fichier.lsb généré par cette commande, en général c'est le même nom du fichier.c. *WkspDir* c'est le nom du répertoire du projet

et *OutDir* c'est le nom du répertoire où se trouvent les deux fichier.obj et fichier.dll générés par Visual C++ qui portent le même noms. Par la suite le fichier.lsb sera généré dans ce répertoire même.

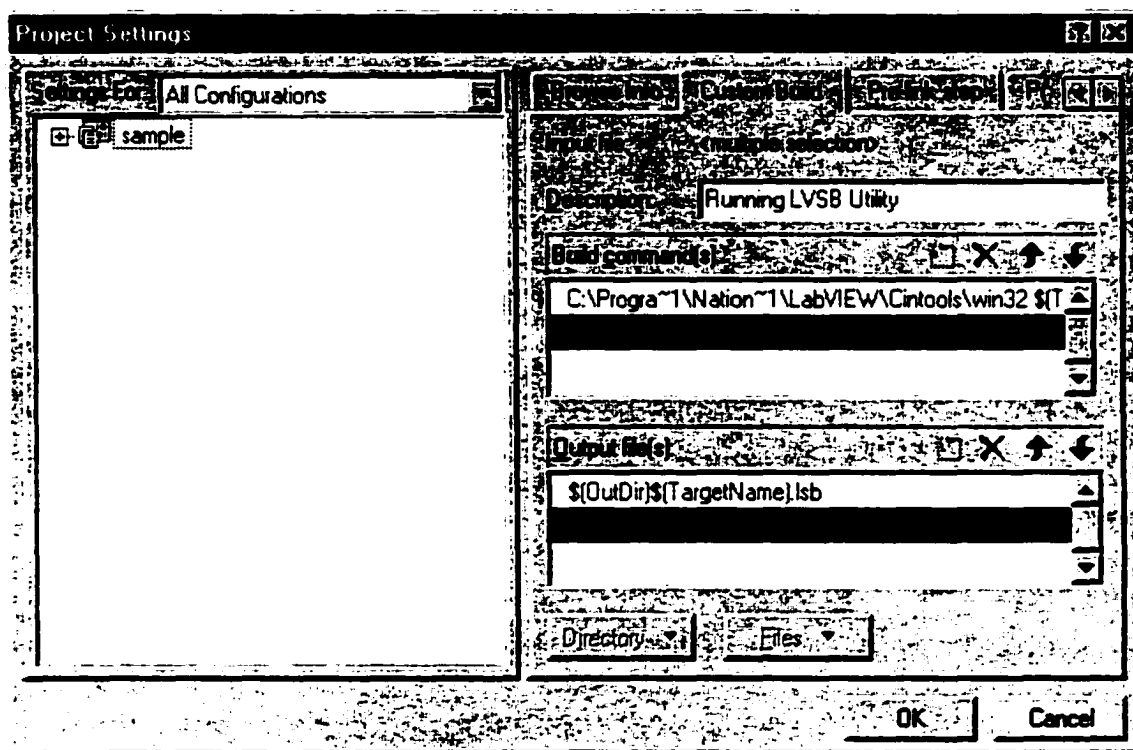


Figure G. 6 : Le chemin d'accès.

e) Ajouter le code C dans le fichier "VI_name.c".

Activer la fenêtre WorkSpace et Choisir "File View".

Double cliquer sur le fichier "VI_name.c" pour l'ouvrir.

Ajouter dans la zone qui indique `/*ENTER YOUR CODE HERE*/` le code C nécessaire pour réaliser l'application.

Il faut que le code se compile bien (sans erreurs) afin qu'il génère le fichier

"VI_name.obj", Vérifier en choisissant Build >> Compile VI_name.c

f) Générer le fichier source " VI_name.lsb ".

Choisir : Build >> Build VI_name.dll

Ceci créera le fichier de VI_name.lsb et le placera dans le répertoire du projet :
VI_name\Debug.

NOTE: Il y aura plusieurs avertissements (*Warnings*) produits pendant le lien, on peut les ignorer.

4- Charger le code source dans LabVIEW

Maintenant qu'on a le code source (VI_name.lsb), on doit l'importer dans notre VI. Pour faire ceci on doit aller de nouveau dans le VI_Name.vi et précisément dans la face arrière (le diagramme). On clique avec le bouton droit de la souris sur la première section du CIN et on choisit "Load Code Resource... ". On Choisit le fichier "VI_name.lsb" de la liste. La flèche qu'on utilise pour exécuter le programme devient blanche, alors le VI_name.vi est prêt à être exécuté. On peut sauvegarder le VI_name.vi, dans ce cas on aura plus besoin du code source original pour exécuter le CIN, sauf si on veut changer le code on peut en tout temps faire une mise à jour et charger un autre code source.

Quel est le format de MS-DOS pour les chemins d'accès : ?

Si les répertoires qui mènent au cintools ont un espace blanc ou sont plus longs que 8 caractères, on doit se référer à ces répertoires en utilisant leur format de MS-DOS. Le format est:

[First_six_letters]~[#]

Là où [First_six_letters] est les six premiers caractères du nom de répertoire avant l'espace blanc, et [#] est le nombre 1. ce nombre peut être plus grand si plusieurs répertoires ont les mêmes six premiers caractères.

Par exemple : si le répertoire de cintools est localisé dans :

C:\Program Files\National Instruments\LabVIEW\Cintools alors le chemin d'accès à ce répertoire est:

C:\Progra~1\Nation~1\LabVIEW\Cintools

On peut trouver les noms du fichier sous format MS-DOS en utilisant une de deux méthodes:

On ouvre Windows Explorer, on clique avec le bouton droit de la souris sur un répertoire et on choisit propriétés puis on regarde le nom sous format MS-DOS.

On fait une recherche sous DOS en utilisant les commandes *Cd* et *dir*.

ANNEXE H

Moteur avec le banc d'essai

Moteur avec le banc d'essai

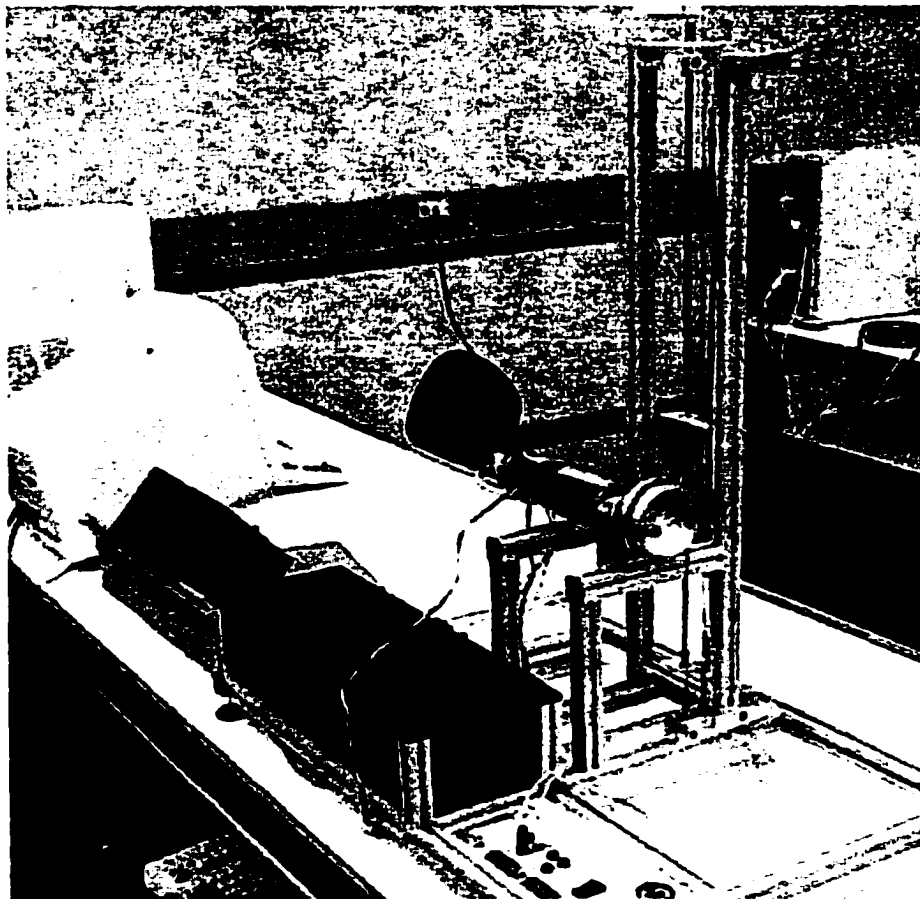


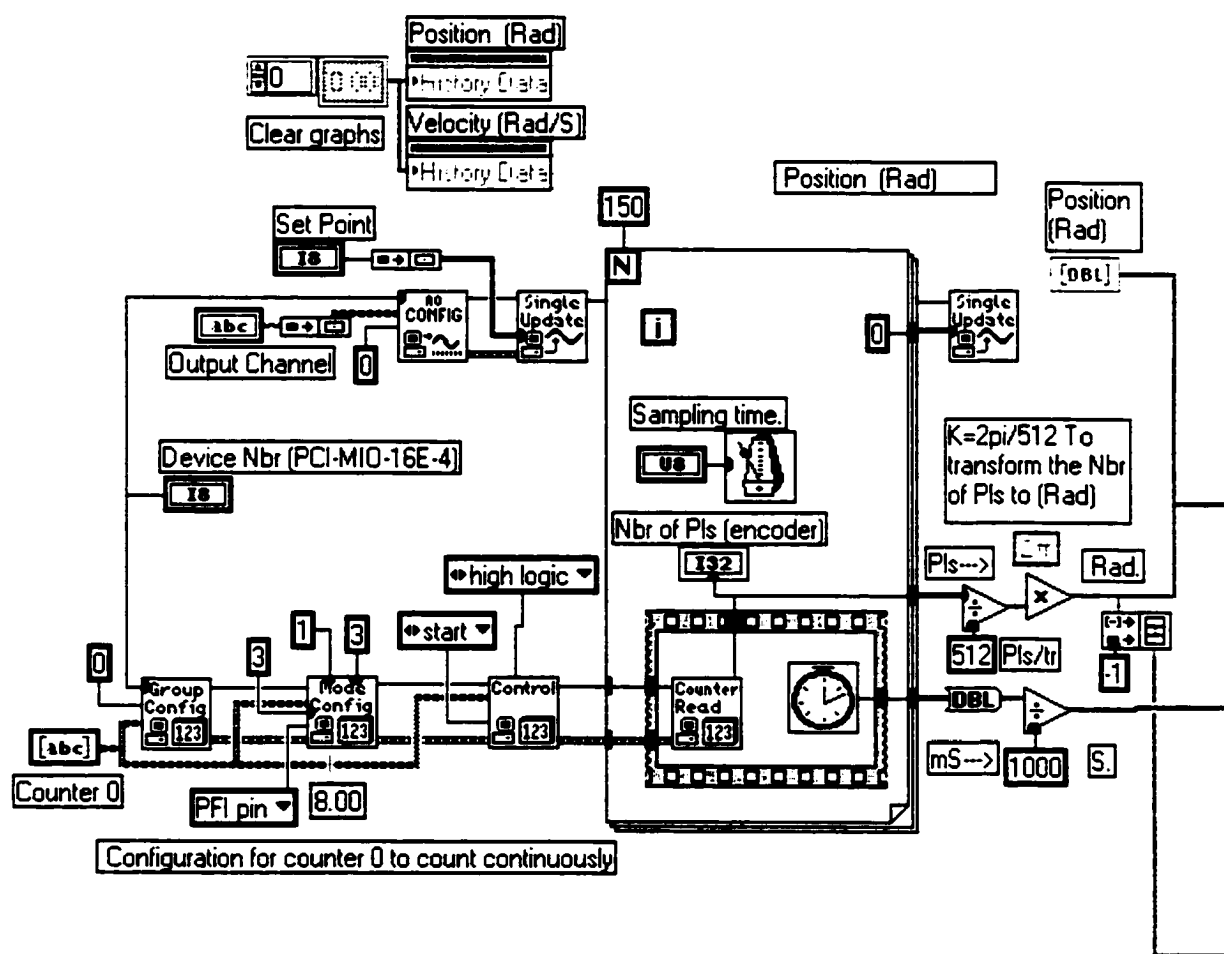
Figure H. 1 : Le banc d'essai avec le moteur.

ANNEXE I

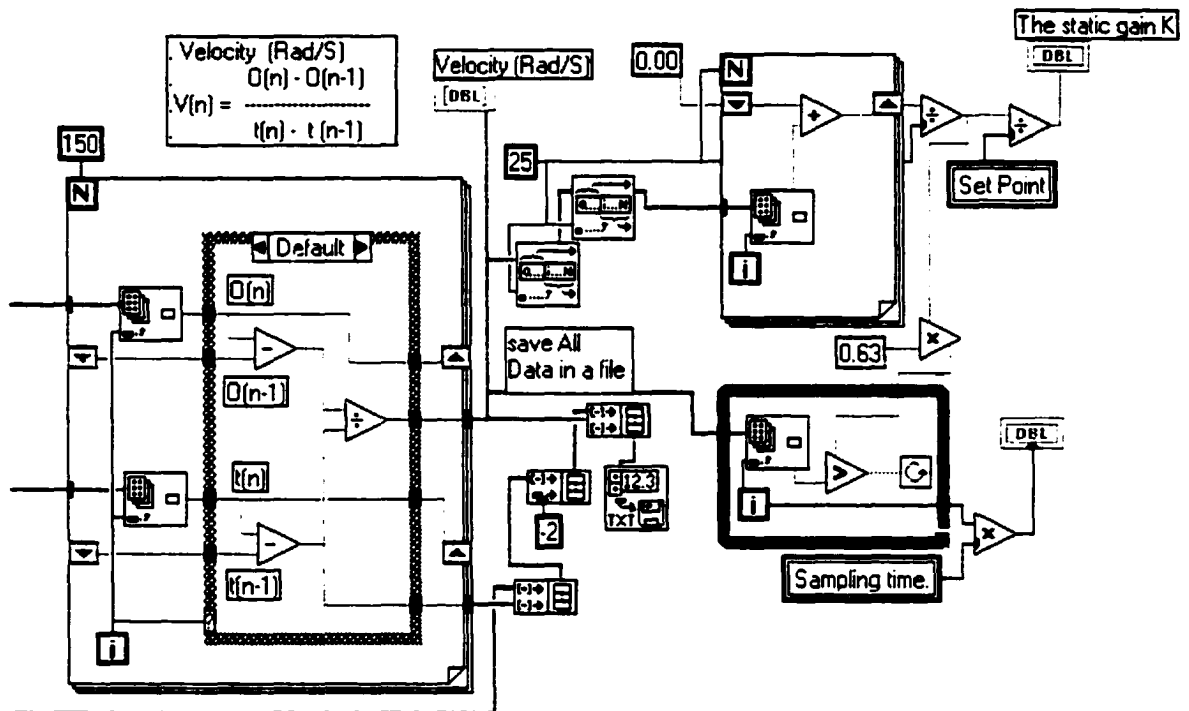
Les diagrammes de LabVIEW

Les diagrammes de LabVIEW

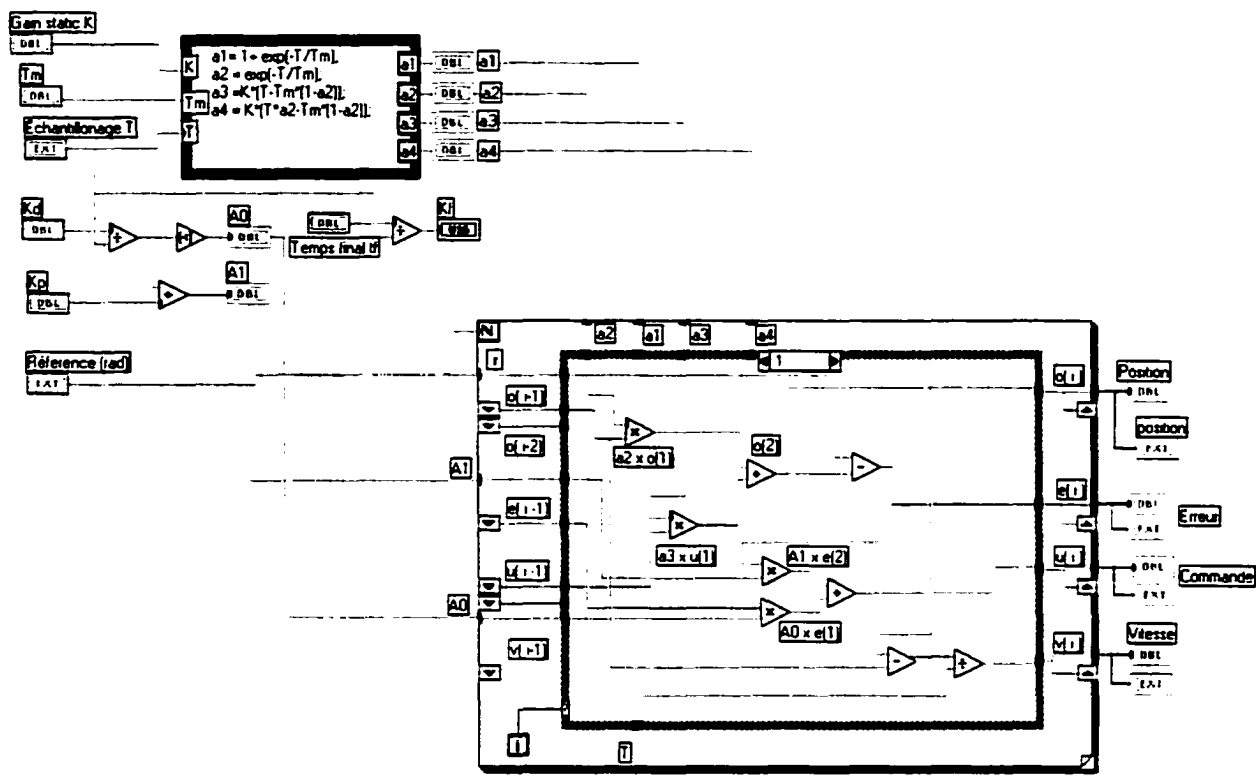
Le programme en boucle ouverte Pour l'identification de paramètres.



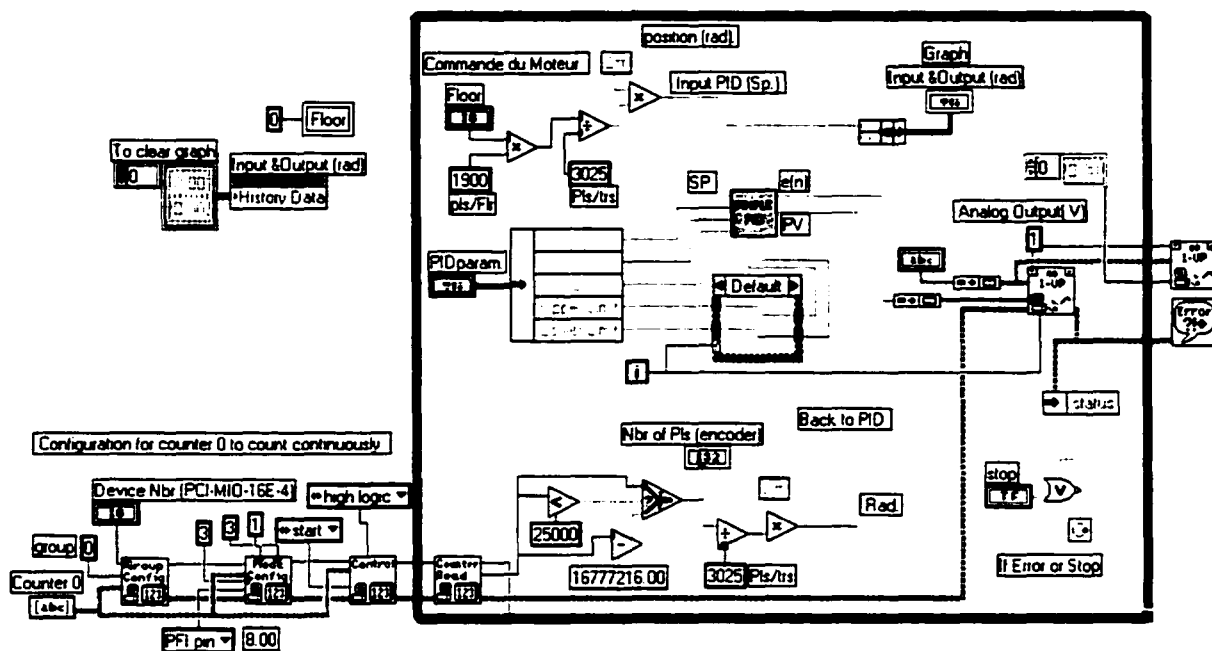
Le programme en boucle ouverte Pour l'identification de paramètres (suite).



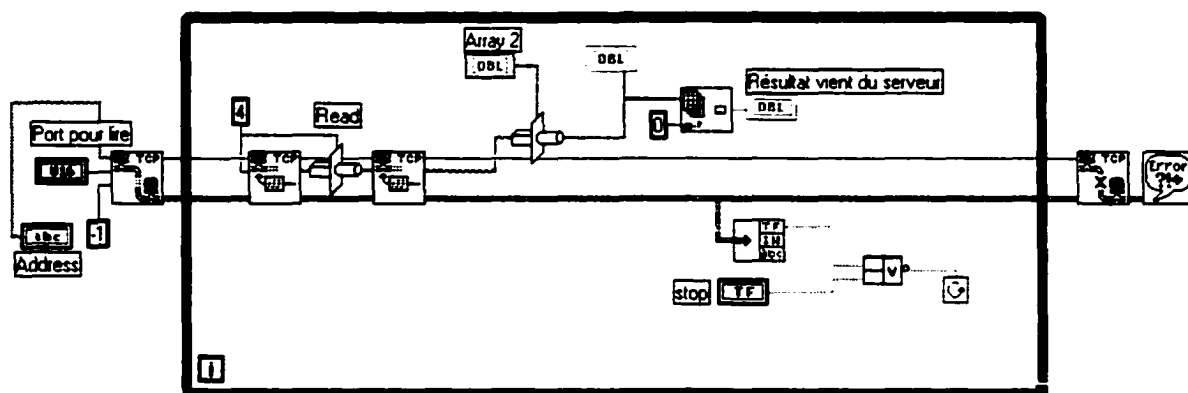
Le programme qui peut simuler le moteur



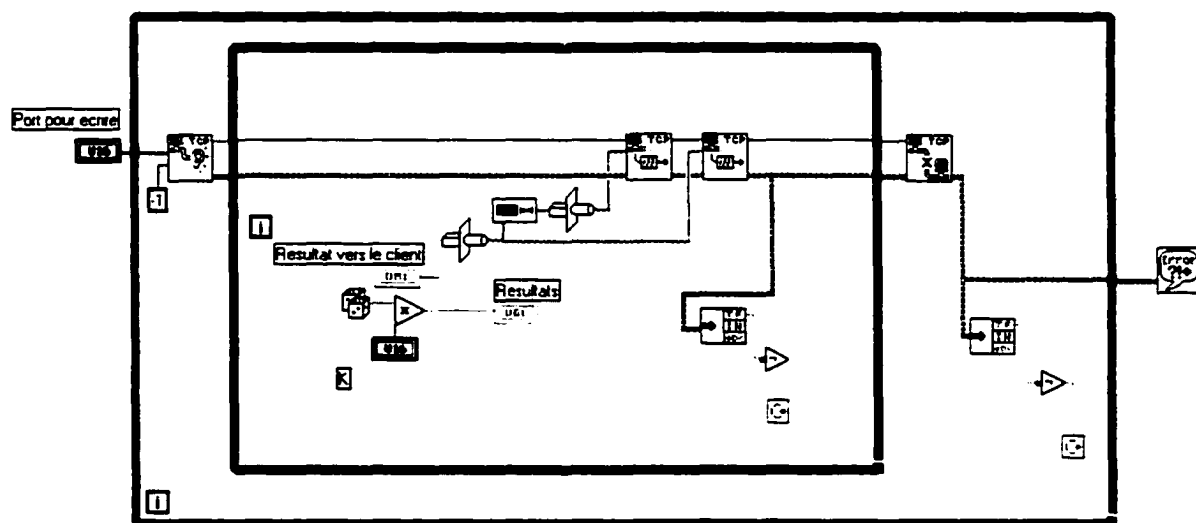
Le programme en boucle fermée avec PID qui permet de contrôler le moteur à travers la carte d'acquisition.



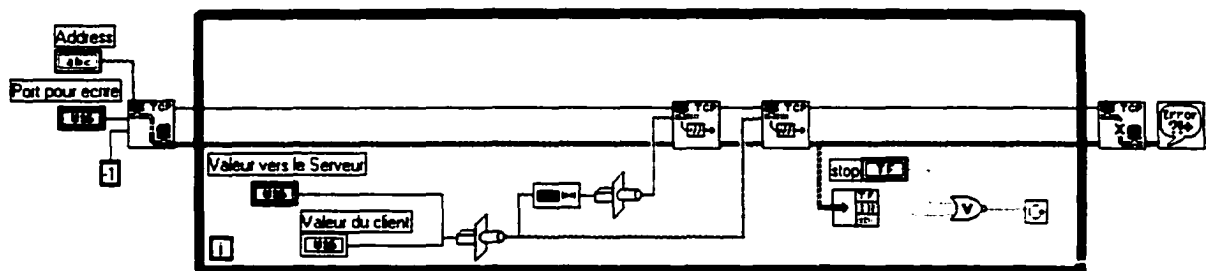
Le client qui reçoit du serveur.



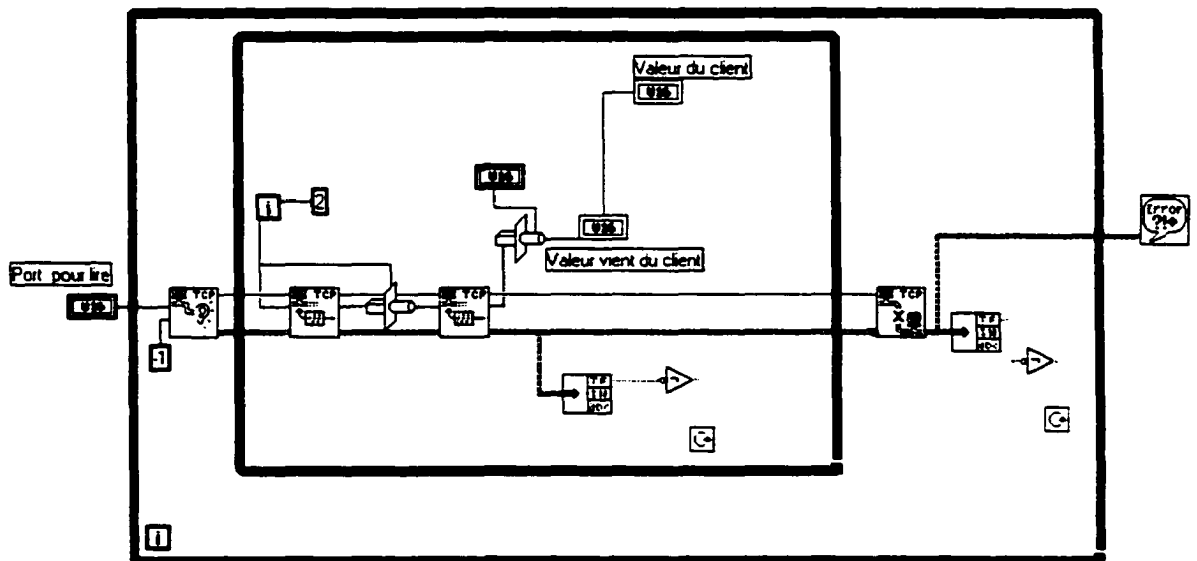
Le serveur qui envoie vers le client.



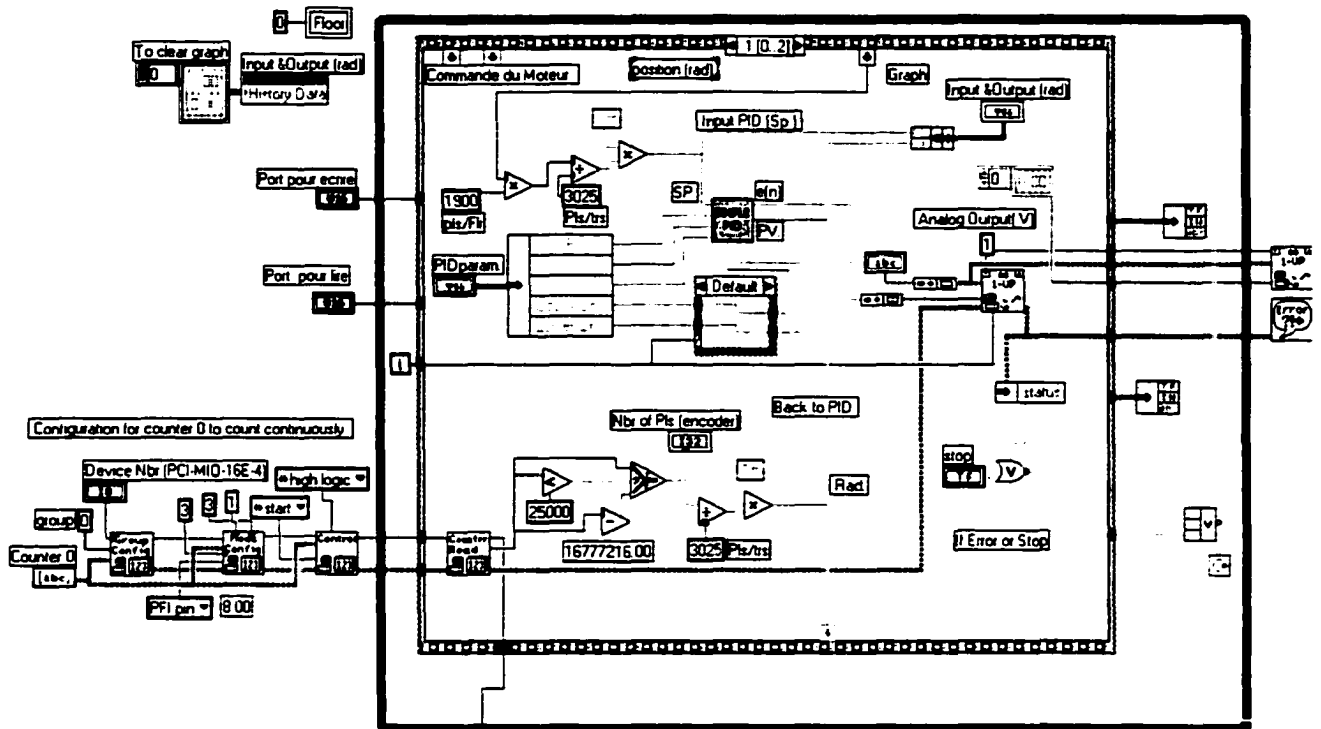
Le client qui envoie vers le serveur.



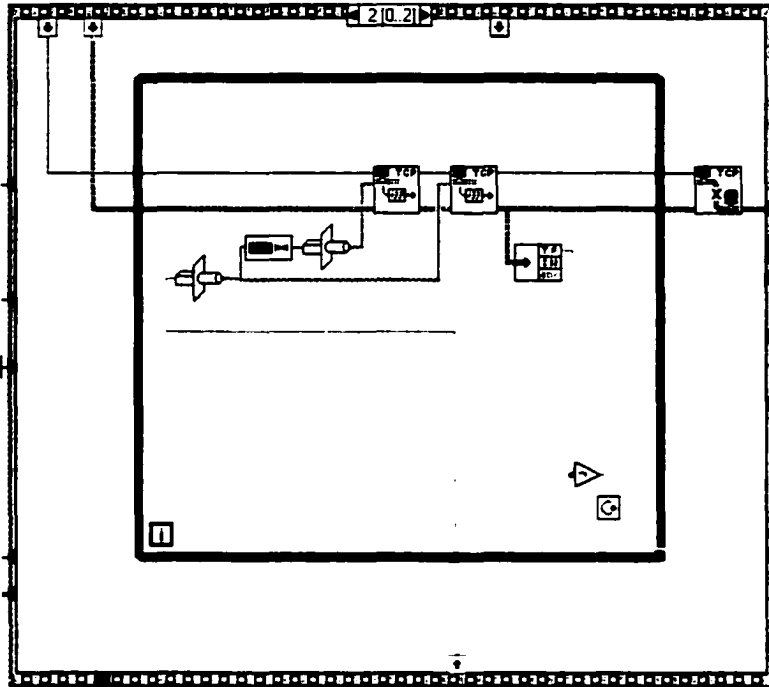
Le serveur qui reçoit du client.



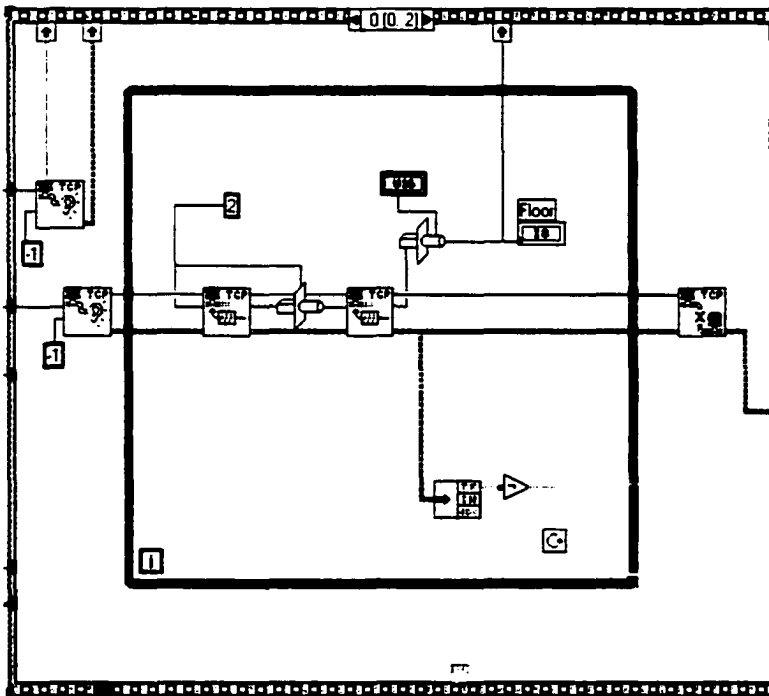
Le serveur qui contrôle le moteur et qui communique avec le client.



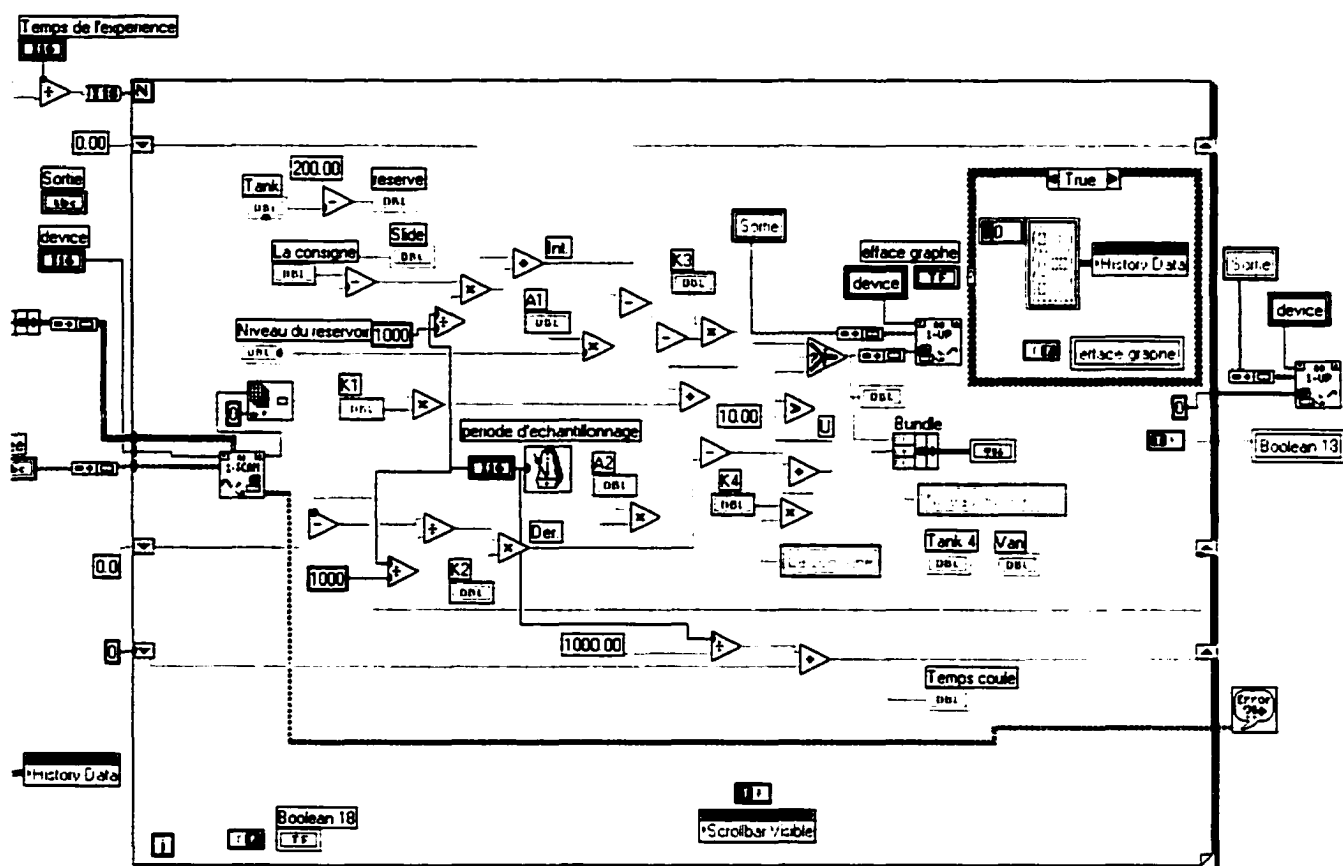
Le serveur qui contrôle le moteur et qui communique avec le client (suite I).



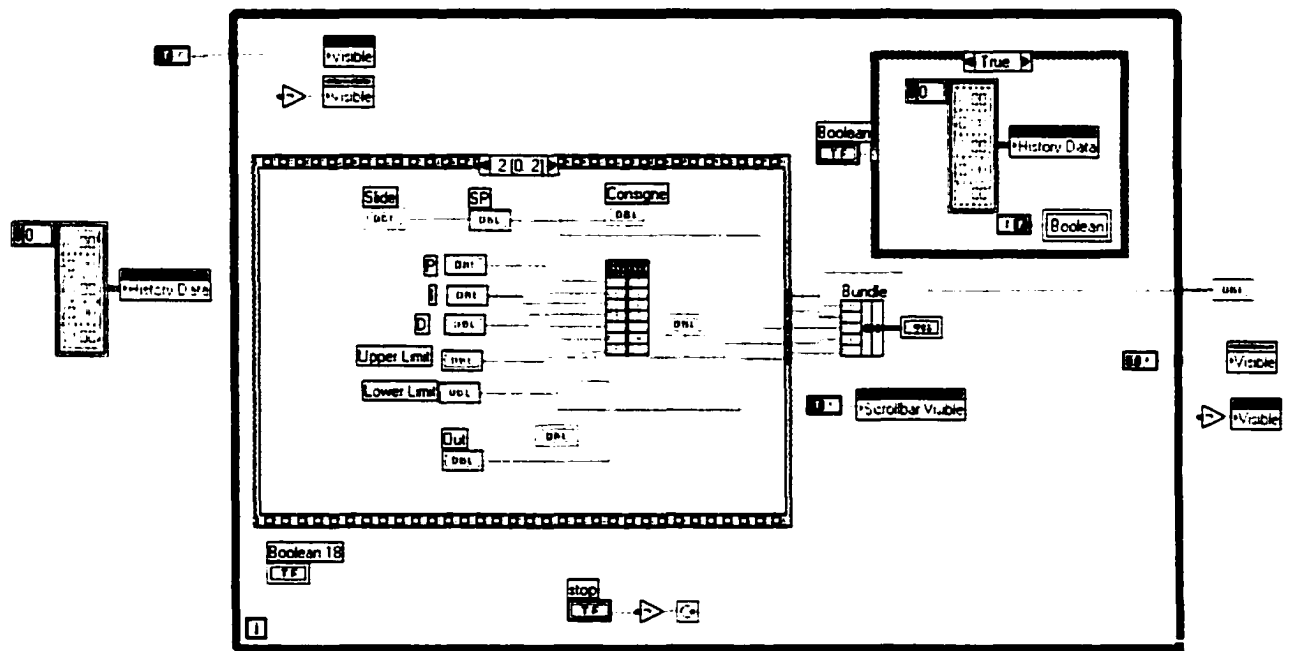
Le serveur qui contrôle le moteur et qui communique avec le client (suite II).



Le programme qui contrôle le simulateur avec la carte d'acquisition.



Le programme qui contrôle le simulateur via le port série en utilisant le CIN.



BIBLIOGRAPHIE

Sylvie Dodeller RFI/Autrans. Le professeur du futur.

<http://www.rfi.fr/kiosque/Mfi/Autrans/vendredi/fiches/fiche20.htm>. Consulté le 29 juin 2001.

Jana Kapounov, Jiri Patocka, Faculté de Pédagogie, Université de Bohême du Sud à Ceske Budejovice. Vers un laboratoire virtuel pour les étudiants en physique

<http://www.epfl.ch/SIC/SA/publications/FI96/fi-sp-96/sp-96-page30.html>. Consulté le 13 février 2001.

Gilles Raillard/Autrans. Bruno Oudet : Un laboratoire virtuel pour la France.

<http://www.rfi.fr/Kiosque/Mfi/Autrans/vendredi/fiches/fice24.htm>. Consulté le 14 mars 2001.

Jeffery Kahn. Virtual Scientific Laboratories Debuting on the Internet

<http://www.lbl.gov/Science-Articles/Archive/remote-microscopy.html>. Consulté le 30 juin 2001.

Roehrig Christof, Jochheim Andreas (Aug 22-27, 1999). Virtual lab for controlling real experiments via internet. IEEE Digital Avionics Systems Conference DASC, Piscataway, NJ, USA.

Munoz C., Rayhan S., Iles T. (Jun 1997). Creating a Virtual Lab for real-time education on the web. Univ of Central Florida Source : IEEE Workshop on Real-Time Systems Education. Los Alamitos, USA.

Laboratoires Virtuels pour L'Éducation en Sciences et Technologie (LVEST).

<http://www.licef.telug.quebec.ca/lvest/index2.htm>. Consulté le 15 janvier 2000.

Olivier ITEANU. Aux armes citoyens internautes !

<http://www.isocfrance.org/PRESSE/isoc/armes.htm>. Consulté le 27 juin 2001.

National Instruments. Support technique. GPIB. <http://www.ni.com>. Consulté le 20 août 2001.

Industrie Canada. <http://www.ic.gc.ca>. Consulté le 15 juin 2001.

Gene F. Franklin, J. David Powell, Abbas Emami-Naeini. Feedback Control of Dynamic system, third edition, 1994.

Well & Al, Prentice Hall. LabVIEW for Everyone. graphical programming, Canada, 1996.

Richard C.Dorf et Robert H.Bishop. Modern Control Systems, 7th Edition, Addison-Wesley Publishing Company 1995.

Pascal Cordan, Sandrine le Ballois. Automatique (Systèmes linéaires et continus), Dunod, Paris, 1998.

Professeure Ouassima Akhrif. Notes de cours : Modélisation et Identification des systèmes dynamiques, ELE 777 Université du Québec, École de Technologie Supérieure. Hiver 2000.

Gérard Blanchet et Jacques Prado. Éléments d'automatique, ellipses/édition 1995.

Toolbook II Instructor, graphical programming, 1997 Asymetrix Systems, Inc.

Jean Châtelain. Machines électriques. Presses polytechnique romandes. Première édition 1983.

Ambrich Chandra. Notes de cours : Machines électriques, ELE-550. Université du Québec, École de Technologie Supérieure.

Guide de démonstration LabVIEW, Édition de mars 1996, référence 321292A-01.

LabVIEW : Guide de prise en main, février 1999, référence 321676C-01.

Amoruso P., Caciotta M., Carbone P., Orsolino Cencelli V. (Mai 19-21, 1997). An implementation of a Parallel Virtual Instrument; *IEEE Instrumentation and measurement Technology*. Conférence Ottawa, Canada.

Saliah H. (Jan. 1996). Formation, coopération et assistance technique à distance médiatisée. *VIèmes Journées Internationales de Technologies*, (CITEF).

Saliah H., Saad M., De la Teja I. (2000). Outils, environnement et méthodologie pour l'encadrement et la formation d'équipes multidisciplinaires en génie travaillant en réseau par le Web.

Observatoire du multimédia de formation (Avril 1999). *Bureau des technologies d'apprentissage (BTA)*.

Goldberg H. (Dec. 2000).What is Virtual Instrumentation? *IEEE Instrumentation & Measurement Magazine*.

L'Industrie du Multimédia. <http://www.multimedia-quebec.com/>. Consulté le 25 juin 2001.